Université de Strasbourg

UNIVERSITÉ DE STRASBOURG

École doctorale Mathématiques, sciences de l'information et de l'ingénieur | ED 269 Université de Strasbourg

ÉCOLE DOCTORALE MATHÉMATIQUES, SCIENCES DE L'INFORMATION ET DE L'INGÉNIEUR – ED269

UMR 7357 / Cerema

THÈSE présentée par :

Guillaume DECOR

soutenue le : **27 septembre 2023**

pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Spécialité : **Traitement du signal et des images**

Reconnaissance des formes pour l'inspection visuelle des tunnels

PRÉSIDENT DU JURY :

M. GRUSSENMEYER Pierre	Professeur, INSA Strasbourg

RAPPORTEURS :

Mme CHAMBON Sylvie	Maîtresse de Conférences, IRIT
M. BERNARDIN Frédéric	IDTPE, Cerema

EXAMINATEUR :

M. BALTAZART Vincent	Chargé de Recherche, UGE

THÈSE co-dirigée par :

M. HEITZ Fabrice	Professeur, Université de Strasbourg
M. CHARBONNIER Pierre	Directeur de Recherche, Cerema

INVITÉS :

Mme DOREAU-MALIOCHE Jeanne	ITPE, CETU
M. FOUCHER Philippe	Chargé de Recherche, Cerema

Cette thèse de doctorat a été financée par le Cerema sous la forme du contrat doctoral n° 2018/0001CD. Elle a été réalisée au sein du Laboratoire de Strasbourg du Cerema Est (Groupe ENDSUM) et de l'équipe IMAGeS du laboratoire ICube de Strasbourg, sous la co-direction de Fabrice Heitz et de Pierre Charbonnier, et l'encadrement de Philippe Foucher.

Dans cette école, on apprenait à désapprendre. Chaque cours servait à remettre en question ce qu'on pensait savoir sur le monde, la vie, les relations humaines, le fonctionnement de la société. A la fin de l'année, un diplôme était remis à ceux qui n'avaient plus de certitudes.

Patrick Baud, *Nanofictions*, éditions Flammarion, 2018.

Remerciements

Je ne peux conclure cette période doctorale sans rendre hommage à toutes les personnes qui, de près ou de loin, ont contribué à la réussite de ce travail.

En premier lieu, je souhaite témoigner ma reconnaissance envers l'ensemble des membres du jury, qui ont accepté d'évaluer mon travail. Merci pour le temps que vous avez consacré à la relecture de mon manuscrit ainsi que pour vos retours constructifs sur ce dernier.

Merci à mes directeurs de thèse, Fabrice et Pierre, ainsi qu'à mon encadrant, Philippe, pour m'avoir accompagné dans ce projet de thèse. Leur expertise m'a été précieuse :

— En matière de reconnaissance des formes, d'une part, pour laquelle Pierre et Philippe (auxquels il convient d'associer Christophe Heinkelé) m'ont permis de monter en compétence dès ma deuxième année de master et, ainsi, d'acquérir les bonnes pratiques de la discipline. Dans un passé plus proche, je tiens à souligner les nombreuses simulations qu'a réalisées Philippe en parallèle de mes travaux et qui, en plus d'étayer certaines de nos hypothèses, ont permis d'enrichir les résultats de la thèse et ont débouché sur une publication au sein d'une conférence internationale. Je souhaite aussi remercier Pierre et Fabrice pour m'avoir fait bénéficier de leur expérience concernant la mise en œuvre des méthodes de régression robuste sur les données LCMS.

— En matière de communication scientifique, d'autre part, où cette expertise s'est avérée inestimable. Outre leur relecture minutieuse de mon manuscrit et leurs conseils avisés sur les présentations et posters qui ont jalonné ces 5 années, le constat est sans appel : nous avons soumis 4 articles durant la thèse, et tous été acceptés. Il ne fait aucun doute que le résultat aurait été différent sans leur aide.

Par-delà ces savoir-faire, j'aimerais aussi remercier mes directeurs de thèse et mon encadrant pour la confiance qu'ils m'ont accordée tout au long de ces années de thèse et pour la grande liberté qu'ils m'ont octroyée lors de mes recherches.

Ma thèse est le fruit d'une collaboration entre le Cerema et le laboratoire ICube. Je tiens à avoir un mot pour l'ensemble des acteurs de ces deux partenaires :

— Côté Cerema, je souhaite remercier l'ensemble des agents de l'antenne strasbourgeoise pour leur sympathie et leur bonne humeur constante. Merci aux membres de l'équipe ENDSUM d'avoir participé aux campagnes d'acquisition de données, données sans lesquelles nombre de mes travaux de thèse n'aurait pu aboutir. Je remercie également Philippe Thirion et Jean-Marc Willer, directeurs successifs de l'agence de Strasbourg, pour leur accueil. Je souhaite bon courage à Marc, doctorant arrivé au printemps 2023, et qui poursuit les travaux du Cerema sur l'inspection des tunnels.

— Un immense merci à l'ensemble de l'équipe IMAGeS du laboratoire ICube pour leur accueil chaleureux. Plus spécifiquement, je souhaite remercier Éléonore et Étienne LQ pour nos intarissables discussions. Merci à Céline, qui a su m'écouter et me conseiller pendant les périodes troubles de ma thèse. Merci à Luc de m'avoir glissé la référence de l'article ICLR sur l'expressivité des couches de *Batch Normalization*. Sans son intervention, qui pourra peut-être lui sembler insignifiante, le chapitre 6 du présent document ne comprendrait que peu de résultats encourageants. Et comment conclure ce paragraphe sans remercier Cyril, mon co-bureau de la première heure ? Expert en configuration des systèmes, hacker dans l'âme et infatigable « loubard » (*dixit* Alexandre), cette thèse lui doit beaucoup.

— Merci enfin aux doctorants issus des autres équipes de recherches et que j'ai eu l'occasion de côtoyer pendant cette thèse, notamment Mélinda, Katia ou encore Nicolas. Votre optimisme et votre bonne humeur ont été d'un grand réconfort.

Merci à mes amis qui m'ont supporté durant ces années. Je tiens tout particulièrement à remercier Morgane et Romain pour leur amitié. Je retiendrai pour longtemps les nombreuses soirées endiablées auxquelles ils m'ont convié. Merci également à Thibault qui, en plus d'être un ami indéfectible de longue date, a directement contribué à ma thèse en participant à la campagne d'acquisition de données.

Parce que la musique occupe une place prépondérante dans ma vie, et que ce secteur a été mis à rude épreuve par la pandémie de covid-19, je souhaite adresser mes remerciements à l'ensemble des musiciens et chef(fe)s d'orchestre avec qui j'ai le plaisir de jouer ces dernières années : les membres de Silverflutes, de Phénix et de l'harmonie municipale de Benfeld. Jouer avec vous a été une véritable bouffée d'oxygène pendant ces années parfois compliquées.

Merci enfin à ma famille et, plus encore, à mes parents et à mon frère, qui m'ont toujours soutenu dans ce projet et qui ont invariablement répondu présents lorsque j'avais besoin d'eux.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

1.1 L'inspection des tunnels en France

Les tunnels représentent une part conséquente et essentielle des infrastructures en France. Ils peuvent être de différentes natures : routiers, ferroviaires ou encore navigables. Tous types confondus, on dénombre 2410 tunnels pour une longueur totale de 1708 km (voir tableau 1.1).

Pour garantir le bon état de service de ces ouvrages et assurer, ainsi, la sécurité des biens et des personnes qui les empruntent, des inspections sont régulièrement organisées sur site. L'objectif de ces inspections est de faire l'inventaire des anomalies présentes sur ces structures afin de pouvoir dresser un diagnostic général sur l'état de l'ouvrage et ainsi de signaler au gestionnaire du tunnel, le cas échéant, la nécessité d'entreprendre des réparations. Les anomalies sont des altérations de l'ouvrage susceptibles d'avoir une incidence sur sa stabilité. Elles peuvent être causées par la situation environnementale du tunnel (hydrographie du site, fréquence et intensité des périodes de gel, sismicité, etc.), de malfaçons lors de la construction (défauts d'étanchéité, techniques de construction non conformes aux standards, etc.) ou encore être liées à l'usage du site (corrosion due à la pollution, collisions véhicules/parois, incendie, etc.). Le CETU (Centre d'Études des Tunnels) recense 47 types d'anomalies possibles [2].

	Routier	Navigable	Ferroviaire	**Total**
Nombre	940	33	1437	2410
Longueur (en km)	398	42	1268	1708

TABLEAU 1.1 – Dénombrement et mesures des tunnels selon leur typologie (sources : [22, 23, 24]).

La figure 1.1 présente quatre exemples d'anomalies : les fissures, les fers apparents, les zones humides et les nids de cailloux. Les fissures sont des pertes de matières dues à des contraintes physiques exercées sur le matériau. Elles sont généralement longilignes et apparaissent souvent plus sombres que le reste

du revêtement. Les fers apparents concernent les structures en béton armé. Elles surviennent, le plus souvent, lorsqu'une infiltration d'eau se produit. Le fer se corrode, prend de plus en plus de volume jusqu'à ce que la matière située devant lui cède. Le fer est alors visible depuis l'extérieur, d'où le nom de ce type d'anomalies. Les zones humides, quant à elles, désignent les zones présentant une présence d'eau, actuelle ou passée. Par réaction chimique ou encore sous l'effet du gel, l'eau peut endommager le tunnel. Les nids de cailloux, enfin, relèvent d'un défaut de fabrication des constructions bétonnées. Pour que le béton soit robuste, il est nécessaire qu'il soit homogène. Un nid de cailloux est la résultante d'une trop grande concentration locale de granulats dans le béton.

(a) Fissure *(source : [25])*

(b) Fers apparents

(c) Zones humides

(d) Nids de cailloux *(source : CETU [2])*

FIGURE 1.1 – Exemples d'anomalies

Une inspection se décompose en trois étapes : un pré-diagnostic visant à déterminer les zones d'intérêt, une phase d'examens sur site pour répertorier l'ensemble des anomalies et l'analyse des données recueillies qui doit permettre d'établir un diagnostic complet pour l'ouvrage [2]. Les phases d'examen sont particulièrement contraignantes puisqu'elles requièrent la fermeture du tunnel. De plus, elles représentent un travail long et fastidieux. Afin de limiter le temps alloué à cette étape, il est donc essentiel de réaliser le pré-diagnostic le plus précis possible. En particulier, avoir une détermination efficace des zones com-

portant des anomalies permet aux opérateurs de concentrer leurs efforts sur ces dernières. La durée du travail sur site s'en voit alors réduite.

1.2 Vers un nouveau paradigme d'inspection

Face à la lourdeur et à la durée de ces inspections, on assiste ces dernières années à une forte volonté de proposer des outils automatiques pour aider l'inspecteur dans son travail. L'objectif est d'adjoindre à la phase de pré-inspection un procédé pour cartographier algorithmiquement les anomalies, et ainsi permettre aux opérateurs de mieux cibler leur inspection. La figure 1.2 présente la différence entre l'approche traditionnelle et celle proposant cette assistance.

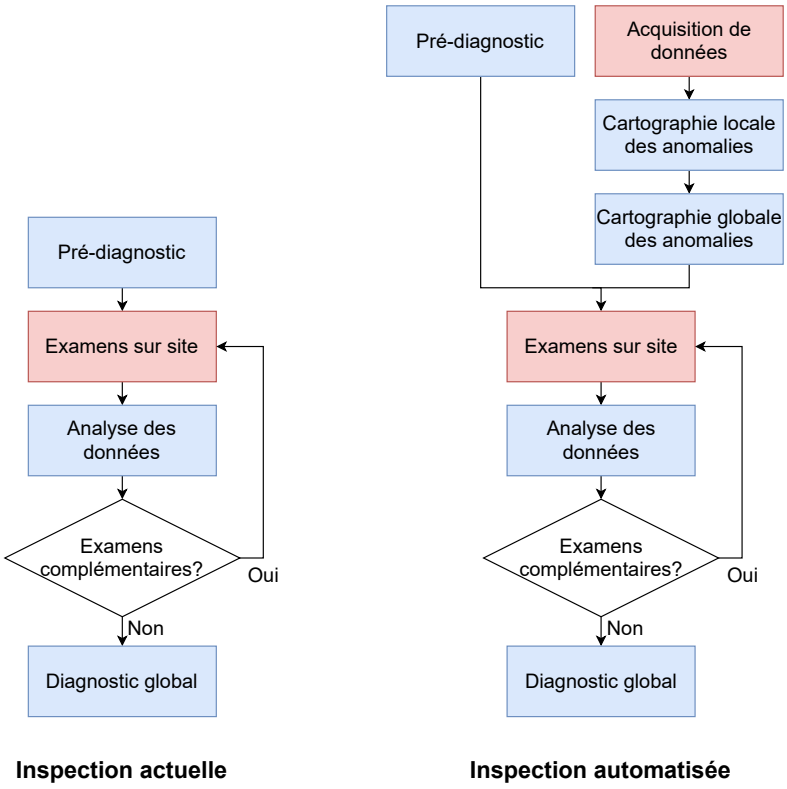


FIGURE 1.2 – Positionnement de l'inspection automatisée que le groupe END-SUM souhaite mettre en place. Une case rouge indique que l'étape correspondante se déroule sur site, impliquant entre autres la fermeture de l'ouvrage, tandis qu'une case bleue marque une étape faiblement contraignante au regard de l'exploitation du tunnel.

Sur ce schéma, la cartographie des anomalies est décomposée en deux sous-tâches : une cartographie locale qui doit permettre de répertorier les anomalies

au sein d'une unité de données (image, relevé laser) et la cartographie globale qui a pour objectif de replacer les cartes locales dans le référentiel global de l'ouvrage. D'un point de vue opérationnel, on cherche à ce que le ciblage permis par la cartographie des anomalies se traduise par un gain de temps lors des examens sur site, gain qui soit suffisamment conséquent pour compenser le temps pris par l'acquisition initiale.

Cette problématique est un axe de recherche important du groupe END-SUM de l'agence strasbourgeoise du Cerema depuis 2010. En 2011, un prototype de système de prises de vue a été développé par ce groupe afin de mener des campagnes d'acquisitions dans le domaine de l'imagerie visible [26]. Destiné à être embarqué sur un véhicule ou sur un bateau, il permet de prendre des photographies de voûtes et de piédroits (voir figure 1.3 pour une explication des termes). Plusieurs travaux ont ensuite été entrepris pour traiter les données acquises. Deux axes principaux sont à l'étude : la détection des anomalies et la reconstruction 3D des ouvrages. Le premier axe doit permettre la cartographie locale des anomalies tandis que le second vise à concevoir une modélisation 3D des ouvrages à partir de données hétérogènes (images, lidar, sonar, etc.). De nombreux stages [27, 28, 29, 30] et une thèse [24] ont permis des avancées sur le second axe. Le premier restait encore exploratoire en 2017, lorsqu'a débuté ma formation en alternance [31] précédant cette thèse. On peut toutefois souligner les travaux réalisés dans le cadre d'un projet de fin d'études en 2015 [32, 33], portant sur la détection d'anomalies sur des images en lumière visible à l'aide de méthodes de forêts aléatoires [34].

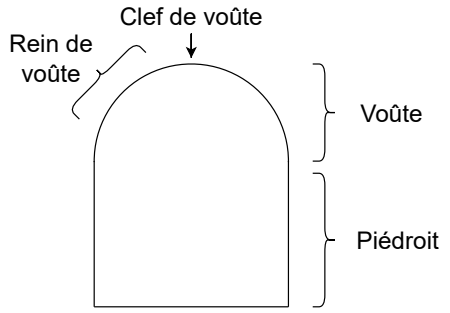


FIGURE 1.3 – Nomenclature des différentes parties du revêtement d'un tunnel.

Depuis 2017, le Cerema collabore avec le CETU et l'entreprise Pavemetrics [35] dans le cadre d'un projet qui vise à concevoir un dispositif à grand rendement d'acquisition de profils 3D du revêtement des tunnels à l'aide de capteurs LCMS (*Laser Crack Measurement System*, voir figure 1.4a) et à en évaluer l'intérêt pour l'inspection des tunnels. Ce capteur projette une ligne laser infrarouge et acquiert, à l'aide d'une caméra, l'image de cette ligne sur une surface. À partir des déformations de la ligne, on en déduit un profil local de profondeur, tandis que l'amplitude du signal réfléchi fournit un profil d'intensité. En agglomérant les profils successifs acquis par le capteur embarqué

sur un véhicule en mouvement, on forme une image d'intensité et de profondeur relative. L'information de profondeur nous intéresse tout particulièrement car une partie des anomalies se traduit par des pertes de matières. Si la surface est modélisée finement, les pertes de matière sont détectables à partir de ces relevés. Ainsi, le capteur LCMS a déjà été mis en œuvre avec succès pour le projet IQRN (Image Qualité du Réseau National) [36], projet dont est ressorti un appareil opérationnel, AIGLE3D, capable d'évaluer l'état du réseau routier en analysant le revêtement à l'aide de ce capteur. En particulier, les capteurs LCMS se sont montrés efficaces pour relever les fissures routières, ce qui a motivé l'étude de l'adaptation de ces derniers au problème, plus difficile, de la détection des anomalies sur les parois des tunnels. Grâce au concours du centre d'étude et de construction de prototypes (CECP) du Cerema-Rouen, un prototype d'acquisition dédié aux tunnels et embarquant un couple de capteurs LCMS a été mis au point [3] (voir figure 1.4b). À ce jour, des acquisitions ont été réalisées dans trois tunnels à l'aide de ce prototype. Pour améliorer la modélisation surfacique des parois, un Projet de Recherche Technologique de l'INSA (PRT) d'une durée de quatre mois a été effectué fin 2019 [37] et un projet de fin d'études a poursuivi ces travaux en 2021 [38].

(a) Capteur LCMS (b) Prototype MALT

FIGURE 1.4 – Capteur LCMS (*Laser Crack Measurement System*) et son intégration au sein du prototype d'acquisition MALT (*Mobile Acquisitions with Lasers in Tunnels*)

1.3 Objectifs de la thèse

La thèse se concentre sur l'axe de la cartographie locale des anomalies à l'aide d'images.

1.3.1 Cartographie locale

Une image I, de taille $W \times H$ et composée de D canaux, peut être vue comme une fonction

$$I : [\![1; W]\!] \times [\![1; H]\!] \times [\![1; D]\!] \to \mathbb{R} \tag{1.1}$$

On note $\mathcal{I}_{W,H,D}$ l'ensemble des images ayant ces dimensions.

Une cartographie locale, pour un type d'anomalie donné et sur une telle image I peut être définie comme un ensemble fini $E = \{E_1, \ldots, E_n\}$ dont chacun des membres est appelé **unité de détection** et appartient à $\mathcal{P}\left([\![1; W]\!] \times [\![1; H]\!]\right)$, où $\mathcal{P}(X)$ désigne l'ensemble des parties de X. Chaque E_i représente alors la détection d'une anomalie au sein de la cartographie, dans le sens où tout $(x, y) \in E_i$ renseigne une anomalie aux coordonnées (x, y) de l'image cartographiée.

Il existe différentes façons de cartographier une image, chacune d'elles contraignant le sous-ensemble $\mathcal{E}$ de $\mathcal{P}([\![1; W]\!] \times [\![1; H]\!])$ dont sont issus les E_i. On définit la notion de boîte englobante discrète par

$$\mathrm{BBox}(x_1, x_2, y_1, y_2) = ([x_1; x_2] \times [y_1; y_2]) \cap ([\![1; W]\!] \times [\![1; H]\!]) \tag{1.2}$$

De plus, la translation d'un ensemble discret E par un vecteur $(\Delta_x, \Delta_y) \in \mathbb{Z}^2$ est notée $T_{(\Delta_x, \Delta_y)}(E)$.

Quadrillage régulier Conceptuellement, la façon la plus simple de cartographier une image est de procéder selon un quadrillage régulier défini en amont. Les unités de détection sont alors constituées de rectangles, tous de dimension $P_w \times P_h$ $(P_w, P_h \in \mathbb{N}^*)$ et disposés selon le quadrillage utilisé. Formellement, on a

$$\mathcal{E}_{\mathrm{reg}} = \left\{ T_{(\alpha_1 x, \alpha_2 y)}(\mathrm{BBox}(0, P_w, 0, P_h)) \mid (x, y) \in \mathbb{Z}^2 \right\} \tag{1.3}$$

où α_1 et α_2 sont les paramètres du quadrillage définissant les écarts minimaux entre deux positions possibles selon chacun des deux axes.

Boîtes englobantes On peut également réaliser une cartographie à l'aide de rectangles de taille et de position arbitraires, c'est-à-dire que l'on a

$$\mathcal{E}_{\mathrm{box}} = \left\{ T_{(x,y)}(\mathrm{BBox}(0, w, 0, h)) \mid (x, y) \in \mathbb{Z}^2, (w, h) \in (\mathbb{N}^*)^2 \right\} \tag{1.4}$$

Chaque unité de détection est alors appelée « boîte englobante ».

Segmentation sémantique Lorsque l'on souhaite une cartographie définie au pixel près, une solution est d'imposer que chaque unité de détection soit réduite à un unique élément, qui est ainsi associée à un unique pixel de l'image cartographiée. Il vient

$$\mathcal{E}_{\mathrm{seg}} = \{\{(x, y)\} \mid (x, y) \in [\![1; W]\!] \times [\![1; H]\!]\} \tag{1.5}$$

On parle de « segmentation sémantique ».

Segmentation d'instances Si l'on souhaite, en plus d'avoir une cartographie de même résolution que l'image, pouvoir distinguer deux anomalies de

même type s'y trouvant, on peut enfin opter pour une cartographie par segmentation sémantique, dite « segmentation d'instances », dans laquelle chaque unité de détection correspond à une instance d'anomalie. On a alors

$$\mathcal{E}_{\text{seg+}} = \mathcal{P}\left([\![1; W]\!] \times [\![1; H]\!]\right) \tag{1.6}$$

Bien que ce ne soit pas systématique, la notion d'instance est généralement entendue au sens de composante connexe.

Ces types de cartographie sont illustrés en figure 1.5.

(a) Quadrillage régulier

(b) Boîtes englobantes

(c) Segmentation sémantique

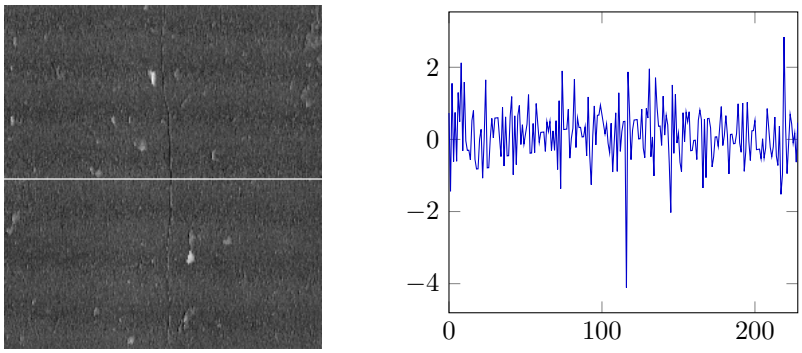
(d) Segmentation d'instances

FIGURE 1.5 – Différents types de cartographie réalisables

S'il est vrai que la littérature propose un grand nombre de contributions pour la cartographie des anomalies dans notre domaine applicatif, elles prennent souvent place dans des configurations simplifiées (*i.e.* faible quantité ou faible représentativité des données en jeu). Or, il a été mis en évidence que de tels biais dans la composition des jeux de données constituaient, d'une manière générale, un frein à leur utilisation en production en dégradant les performances des modèles qui en sont issus [39]. Un des objectifs de ce travail est donc d'évaluer l'ampleur de certains de ces biais et d'y apporter des solutions.

1.3.2 Évaluation de l'apport de la 3D

Outre la cartographie des anomalies en elle-même, on s'intéresse également à l'apport éventuel que peut représenter l'information de profondeur des relevés LCMS pour la construction de cette dernière. En effet, de nombreuses anomalies se traduisent par une variation de la profondeur, en particulier lorsque l'anomalie induit une perte de matière (fissures, fers apparents, etc.). Nous avons, par exemple, pu constater que les fissures étaient visibles dans la composante de profondeur (voir figure 1.6).



(a) Exemple de fissure (b) Profil mesuré le long de la ligne blanche

FIGURE 1.6 – Illustration du potentiel de l'information de profondeur des relevés LCMS. La sous-figure (a) est la carte de profondeur d'une zone comprenant une fissure et (b) représente la profondeur du profil (en millimètres) relevé le long de la ligne horizontale blanche tracée sur l'image d'exemple. On peut y voir que lorsque la ligne blanche est au-dessus de la fissure, le profil admet une profondeur significativement plus basse que pour le restant des valeurs, révélant ainsi la présence de la fissure.

1.4 Verrous scientifiques et opérationnels

Variabilité visuelle intra-domaine Une première difficulté réside dans la diversité d'aspect que présentent les anomalies. Dans l'exemple présenté en figure 1.7, plusieurs fers apparents sont visibles et attestent de cette variabilité visuelle. En effet, on peut y observer que les armatures métalliques peuvent être orientées selon différents angles, partiellement occultées et de largeur variable. De plus, l'un des fers apparents présente, à l'inverse des autres, deux armatures visibles perpendiculaires. Cette variabilité d'aspect se constate aussi pour les pertes de matière les entourant, ces pertes ayant de surcroit des formes générales beaucoup moins contraintes que les armatures. En plus de la variabilité des anomalies en elles-mêmes, il y a également une diversité dans leurs manifestations indirectes, c'est-à-dire l'ensemble des phénomènes provoqués par une anomalie et qui sont visibles dans l'image. Ces manifestations sont importantes à relever car, bien qu'elles ne constituent pas nécessairement des anomalies en

tant que telles, elles peuvent en trahir la présence. Dans notre exemple, c'est spécifiquement le cas pour les dépôts de rouille. Ces dépôts admettent des largeurs et des teintes diverses.

Il ressort de ces observations qu'il est difficile de définir un critère formel ou algorithmique pour déterminer si un objet présent dans une image est une anomalie.

FIGURE 1.7 – Illustration de la variabilité d'aspect caractérisant les anomalies au sein d'un même site.

Biais de domaine Une autre difficulté est induite par la faible représentativité des données disponibles au sein d'un unique site. Ce point est particulièrement problématique dans notre cas, puisque les données acquises au sein de différents tunnels ont tendance à avoir des apparences hétérogènes. Cette variabilité s'explique aussi bien par des différences intrinsèques (nature des matériaux, type d'équipement présent, etc) qu'extrinsèques (conditions d'illumination du revêtement, angle et distance entre la paroi et le capteur, etc). Ainsi, une méthode conçue pour un ouvrage risque de témoigner de faibles performances sur un autre. Cette variabilité est illustrée à la figure 1.8, qui donne à voir quatre sites distincts, dont le détail est présenté en chapitre 2 :

a) Un tunnel piéton

b) Un bâtiment universitaire

c) Un pont

d) Le tunnel routier de Rive-de-Gier

Il apparaît que les surfaces bétonnées du bâtiment universitaire (1.8b) présentent des traces de moisissures absentes du tunnel piéton (1.8a). De plus, les fers apparents de l'exemple issu du pont (1.8c) sont de taille variable, présentent une coloration tirant vers l'orange du fait de la rouille et sont régulièrement pris en très gros plan. Enfin, on peut noter que les pertes de matières entourant les fers apparents au sein du tunnel routier (1.8d) sont généralement plus claires que le fond, ce qui ne survient pas dans les autres jeux de données.

(a) Tunnel piéton

(b) Bâtiment universitaire

(c) Pont *(source : [40])*

(d) Tunnel routier de Rive-de-Gier

FIGURE 1.8 – Illustration de la variabilité d'aspect caractérisant les anomalies se trouvant sur des ouvrages différents. Quatre exemples de fers apparents issus de quatre sites différents sont présentés.

Pour illustrer le biais de domaine, nous avons réalisé une visualisation de l'ensemble des images photographiques utilisées dans cette thèse (voir figure 1.9) à l'aide de l'algorithme t-SNE [1]. Dans cette représentation, chaque point correspond à une image. Pour des raisons techniques, ces dernières sont redimensionnées de sorte que leurs résolutions soient de 256×256, après quoi est réalisée une analyse par composantes principales sur ces images réduites,

visant à plonger chacune d'elles dans $\mathbb{R}^{100}$. L'algorithme de visualisation est alors appliqué sur les vecteurs ainsi formés.

On peut y voir que les sources constituent des amas essentiellement cohérents et globalement distincts les uns des autres, en particulier pour le tunnel routier de Rive-de-Gier qui est nettement différencié des autres sources et dont le nuage de points présente, de surcroît, un caractère multi-centrique, chaque amas représentant une région spécifique du revêtement (piédroit, différentes hauteurs du rein de voûte et clef de voûte).

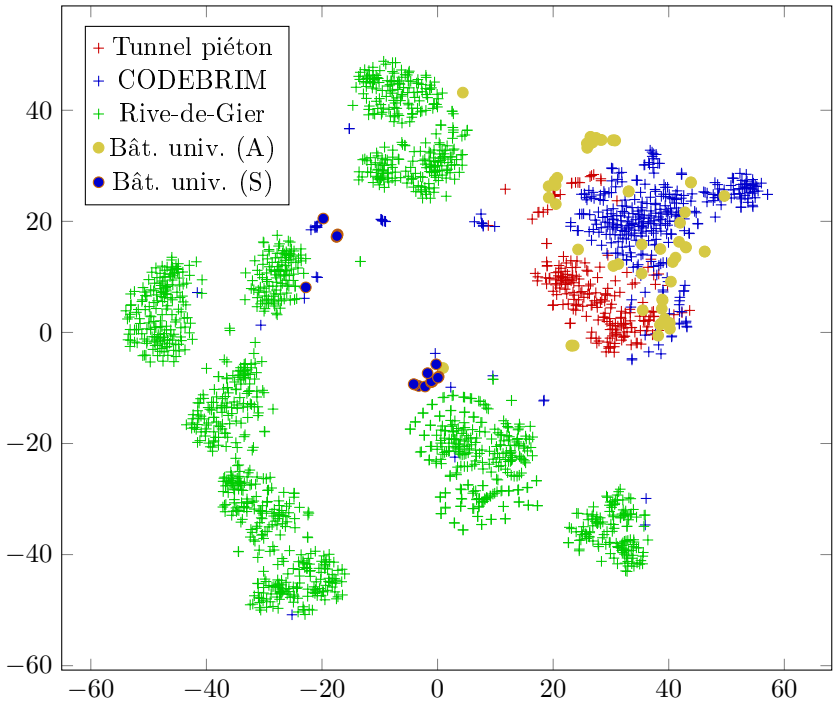
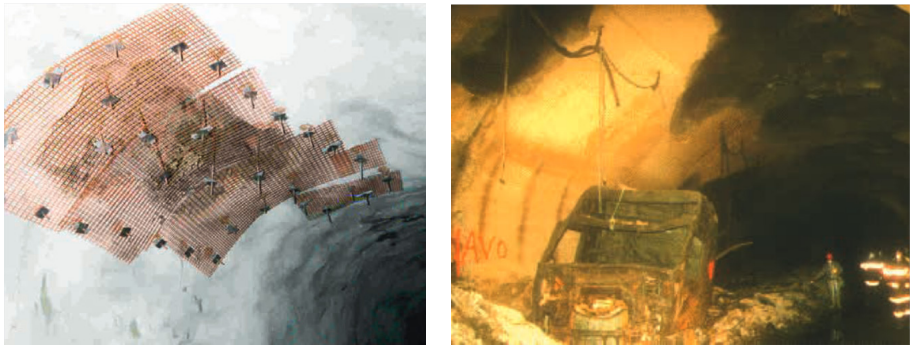


FIGURE 1.9 – Visualisation via t-SNE [1] des images photographiques regroupées par sources de données (les différentes sources sont présentées dans le chapitre 2).

Rareté de certains types d'anomalie La rareté de certaines anomalies constitue également un verrou important. Il est, en effet, complexe de mettre au point une méthode de détection pour un certain type d'anomalie dès lors que peu d'exemples de cette anomalie sont disponibles. La situation est d'autant plus critique que certaines anomalies rares sont particulièrement sévères. La figure 1.10 en donne deux exemples en montrant les conséquences d'une rupture de voûte (effondrement partiel de la voûte) et d'un incendie provoqué par un accident de poids lourd.



(a) Rupture de voûte (b) Incendie

FIGURE 1.10 – Anomalies rares et sévères *(source : CETU [2])*

Recouvrement entre classes Il existe enfin une difficulté due au recouvrement entre classes. En effet, il arrive que l'image n'apporte pas suffisamment d'informations pour déterminer la nature de certains objets représentés, y compris pour un expert. Le plus souvent, la confusion porte sur l'appartenance à une catégorie d'anomalie ou à la classe saine. Par exemple, certaines fissures sont très fines et il n'est pas toujours aisé de déterminer avec précision leur tracé dans l'image. Un autre exemple, plus problématique encore, concerne les zones humides et est détaillé dans le chapitre 2, section 2.1.2.

1.5 Approche générale

Pour répondre à ces verrous, nous mettons en place une stratégie basée sur l'apprentissage par l'exemple et, plus précisément, l'apprentissage profond [41].

1.5.1 Apprentissage supervisé

L'apprentissage supervisé repose sur l'utilisation d'un modèle de cartographie et d'une banque d'exemples annotés. On note

$$\mathcal{D} = \{I_1, \ldots, I_p\} \subset \mathcal{I}_{W,H,D} \tag{1.7}$$

l'ensemble des images et $\Gamma : \mathcal{I}_{W,H,D} \to \{0;1\}^{\mathcal{P}(\mathcal{E}_X)}$ est la fonction qui associe, à chaque exemple, la cartographie attendue pour ce dernier. Une telle cartographie est appelée **vérité terrain**

Un **modèle de cartographie** est une application

$$f_\theta : \mathcal{I}_{W,H,D} \to [0;1]^{\mathcal{P}(\mathcal{E}_X)} \tag{1.8}$$

où $\theta = \{\theta_1; \ldots; \theta_m\}$ est l'ensemble des paramètres du modèle, $X \in \{\text{reg; box; seg; seg+}\}$ désigne le type de cartographie considéré et $[0;1]^{\mathcal{P}(\mathcal{E}_X)}$ l'ensemble des applications $\mathcal{P}(\mathcal{E}_X) \to [0;1]$. En d'autres termes, pour toute image I, $f_\theta(I)$ est

une application qui, à chaque unité de détection possible, associe une probabilité. Cette probabilité est proche de 0 si le modèle considère que l'unité de détection correspondante n'est pas une anomalie et s'approche de 1 dans le cas contraire.

Pour déterminer les paramètres du modèle, on cherche à minimiser une fonction de **coût**, qui est une fonction de la forme

$$\mathcal{L}\colon [0;1]^{\mathcal{P}(\mathcal{E}_X)} \times \{0;1\}^{\mathcal{P}(\mathcal{E}_X)} \to \mathbb{R} \tag{1.9}$$

Cette fonction doit, de plus, être différentiable selon son premier paramètre. Dans le cadre de cette thèse, nous utilisons principalement l'entropie croisée binaire et pondérée comme fonction de coût. Cette fonction est notée $\mathcal{L}_{ce}$ et est définie par

$$\begin{aligned}\mathcal{L}_{ce}(\pi_0, \Gamma_0) = -\sum_{E\in\mathcal{P}(\mathcal{E}_X)} \Big[& \mathfrak{w}_0 \Gamma_0(E) \log(\pi_0(E)) + \\ & \mathfrak{w}_1 (1-\Gamma_0(E)) \log(1-\pi_0(E)) \Big]\end{aligned} \tag{1.10}$$

où $\mathfrak{w}_i = \frac{1}{\sum_{j=1}^{p} \#\{\Gamma_0(I_j)^{-1}(\{i\})\}}$ est l'inverse de la fréquence d'apparition de la classe i au sein de la banque de données ($i = 0$ pour un revêtement sain et $i = 1$ pour l'anomalie considérée). On peut démontrer que $\mathcal{L}_{ce}(\pi_0, \Gamma_0)$ atteint son minimum lorsque $\pi_0 = \Gamma_0$. Ainsi, minimiser $\mathcal{L}_{ce}(\pi_0, \Gamma_0)$ revient à faire tendre π_0 vers Γ_0, et donc à faire coïncider la prédiction avec la vérité terrain.

Les valeurs de θ sont alors déterminées itérativement par l'algorithme de descente de gradient [42] ou l'une de ses variantes. À partir d'une telle banque d'exemples et d'un jeu de paramètres initial $\theta^{(0)}$, on construit une suite $\left(\theta^{(n)}\right)_{n\in\mathbb{N}}$ définie récursivement par

$$\theta^{(k+1)} = \theta^{(k)} - \mu \frac{1}{\#\mathcal{D}} \sum_{I\in\mathcal{D}} \nabla\mathcal{L}\left(f_{\theta^{(k)}}(I), \Gamma(I)\right) \tag{1.11}$$

où $\mu \in \mathbb{R}_+$ est un hyper-paramètre, appelé **pas d'apprentissage**, contrôlant l'amplitude accordée au gradient dans l'ajustement des paramètres et $\nabla\mathcal{L}$ est le gradient de $\mathcal{L}$. Le passage de $\theta^{(k)}$ à $\theta^{(k+1)}$ implique que l'intégralité des exemples d'apprentissage a été présentée au modèle. On dit qu'on réalise une **époque**.

Si $\#\theta$ est suffisamment grand par rapport à $\#\mathcal{D}$, il existe un risque que le modèle résultant d'un apprentissage supervisé soit sur-spécialisé. On parle alors de **sur-apprentissage**. Pour mesurer l'ampleur ce phénomène, on divise $\mathcal{D}$ en deux sous-ensembles disjoints, le premier étant utilisé lors de la phase d'apprentissage tandis que le second sert à évaluer le modèle appris. On parle de jeux d'**apprentissage** et de **test**. Il arrive qu'un troisième jeu, également disjoint des deux autres, soit introduit afin de déterminer la valeur de certains hyper-paramètres. Ce jeu est appelé jeu de **validation**.

1.5.2 Stratégie proposée

Idéalement, on souhaite obtenir, pour chaque type d'anomalie, un modèle de cartographie universel, c'est-à-dire directement utilisable dès qu'un nouvel ouvrage est à analyser. Néanmoins, compte tenu du fort biais de domaine, cela supposerait de réaliser l'acquisition et l'annotation d'un grand nombre de tunnels, de l'ordre de plusieurs dizaines. Or, nous sommes naturellement contraints par les moyens matériels et humains de l'équipe ENDSUM. Il nous est ainsi inenvisageable de procéder à une telle campagne dans un délai raisonnable.

À l'inverse, concevoir une méthode de cartographie pour chaque tunnel est peu intéressant sur le plan opérationnel. En effet, cette approche implique d'annoter de larges portions de chacun de ces tunnels. L'utilisation de méthodes de cartographie automatiques ne présenterait alors plus grand intérêt en comparaison d'une annotation manuelle intégrale des ouvrages considérés.

Pour répondre à cette difficulté, nous optons pour une approche hybride. Les modèles de cartographies sont appris sur un nombre réduit d'ouvrages, alors intégralement annotés. Puis, dès qu'un tunnel nouvellement acquis est à analyser, ces modèles sont adaptés à ces ouvrages par le biais de stratégies dites d'**adaptation de domaine**. Ces stratégies consistent à ajuster les paramètres d'un modèle appris sur un domaine (en l'occurrence, un tunnel) pour le rendre plus performant sur un autre, et ce en utilisant peu d'annotations pour ce nouveau domaine, voire aucune pour certaines méthodes. Ainsi, même si ce processus est possiblement coûteux en termes de temps de calcul, le besoin d'intervention humaine est limité.

1.6 Contributions

Dans cette thèse, nous mettons en œuvre une chaîne de traitements complète pour la cartographie des anomalies. Deux types de cartographies sont considérées, par quadrillage régulier et par segmentation sémantique, sur deux types de données, des images photographiques et des relevés LCMS.

Images photographiques Pour la cartographie par quadrillage régulier, nous poursuivons les travaux entrepris au sein du groupe ENDSUM. Sur une banque de données réduite et composée d'images extraites d'un même tunnel, nous comparons les approches d'apprentissage profond avec d'autres méthodes d'apprentissage supervisé. Comme les jeux de données utilisés sont expurgés des exemples difficiles du tunnel, et biaisent donc favorablement les résultats quantitatifs, nous évaluons qualitativement ces deux approches sur l'ensemble des images de l'ouvrage.

Quant à la cartographie par segmentation sémantique, nous la mettons en œuvre dans le cadre de la reconnaissance des fers apparents sur plusieurs sites. À cette fin, nous avons procédé à l'acquisition de deux ouvrages ainsi qu'à la récupération d'une banque de données constituée par d'autres chercheurs. L'ensemble des données, qu'il s'agisse d'acquisitions antérieures à la thèse, de nouvelles acquisitions, ou de données mises à disposition par la communauté

scientifique, est annoté grâce à un outil d'annotation réalisé par mes soins. Nous développons alors une approche multi-échelle pour gérer efficacement la forte variabilité de dimension que présentent les fers apparents au sein des images. L'apprentissage et l'évaluation sont réalisés sur l'intégralité des exemples des ouvrages, là où une partie des expérimentations présentées dans la littérature opèrent sur des bases restreintes, où seules les données comportant des anomalies sont utilisées. Ce faisant, nous obtenons une meilleure estimation des performances des modèles en conditions opérationnelles. De plus, pour cette même tâche, nous quantifions le biais de domaine à travers une évaluation croisée (apprentissage sur un domaine, évaluation sur d'autres) et évaluons différentes stratégies d'adaptation de domaine : une stratégie non supervisée, d'une part, et une autre faiblement supervisée, d'autre part, pour laquelle un opérateur doit manuellement annoter une unique image du domaine cible.

Relevés LCMS Pour les relevés LCMS, nous évaluons, pour les deux types de cartographies mises en œuvre, l'apport respectif des modalités d'intensité et de profondeur pour la reconnaissance de fissures. De manière analogue aux images photographiques, nous avons procédé à l'acquisition et à l'annotation des données. Un logiciel d'annotation de fissures a également été réalisé et permet de relever de façon semi-automatique le tracé de ces dernières. Également, la problématique de la gestion de la profondeur (*cf.* chapitre 2, section 2.2.3) est approchée à travers différents traitements de cette composante de profondeur. En particulier, nous mesurons l'influence de ces traitements sur les modèles entraînés sur les données qui en sont issues.

Concernant la cartographie par quadrillage régulier, nous implémentons et comparons deux approches. La première emploie un unique quadrillage aligné sur les bords haut et gauche de l'image tandis que la seconde a recours à un quadrillage additionnel, dont l'origine est décalée spatialement par rapport au premier.

Nous évaluons également une méthode de cartographie par segmentation sémantique. En plus des indicateurs courants, nous calculons d'autres métriques issues de la littérature et tenant compte de la faible épaisseur que représentent les fissures au sein des images.

1.7 Organisation du manuscrit

Le chapitre 2 présente l'ensemble des données annotées que nous avons utilisé dans le cadre de nos expérimentations, qu'il s'agisse de photographies en imagerie visible ou encore de relevés laser acquis avec le capteur LCMS. La composition des jeux de données ainsi que la fréquence et la nature des anomalies recherchées y sont détaillées.

Dans le chapitre 3, nous revenons sur l'approche générale qui a été retenue pour cette thèse ainsi que sur les métriques d'évaluation utilisées. C'est, en particulier, dans ce chapitre que nous présentons les fondamentaux de l'ap-

prentissage profond ainsi que les modèles dont nous nous sommes servis dans nos travaux.

Nous passons en revue les principales approches qui ont été appliquées dans notre champ applicatif à travers un état de l'art au sein du chapitre 4. Un accent particulier est mis sur les différents jeux de données employés, par rapport auxquels nous nous positionnons.

Dans le chapitre 5, nous mettons en œuvre des méthodes de cartographie afin de répondre au verrou de variabilité visuelle intra-domaine. Tout d'abord, nous illustrons les limites des configurations restreintes issues de précédentes expérimentations de l'équipe ENDSUM. Nous reprenons alors deux approches de cartographie, la première par quadrillage régulier et la seconde par segmentation sémantique, pour les implémenter et les évaluer dans un contexte réaliste sur les deux types de données dont nous disposons.

Le verrou de variabilité visuelle inter-domaine est étudié dans le chapitre 6. Dans un premier temps, nous quantifions l'influence du biais de domaine à travers une étude multi-sites (apprentissage sur un ouvrage, évaluation sur d'autres). Différentes méthodes d'adaptation de domaine sont ensuite évaluées. Une méthode non supervisée, dans un premier temps. Dans un second temps, une fraction des données du tunnel ciblé est labellisée pour pouvoir être utilisée au sein d'une autre approche.

Nous présentons la synthèse de nos travaux dans le chapitre 7, chapitre qui clôt ce manuscrit et dans lequel nous donnons plusieurs perspectives à cette thèse.

Chapitre 2

Données d'infrastructures pour le relevé d'anomalies

Dans ce chapitre, nous décrivons les données utilisées pour nos expérimentations. Nous avons travaillé avec deux types de données : des images photographiques et des relevés laser obtenus par les capteurs LCMS.

Sommaire

2.1 Images photographiques

2.1.1 Généralités

Les images photographiques utilisées dans nos expérimentations sont issues de sources variées. Bien que la thèse porte sur la cartographie des anomalies au sein des tunnels, nous avons décidé d'étendre ces sources aux prises de vue d'autres constructions présentant les mêmes types d'anomalies, telles que les ponts. En plus de représenter une banque d'images plus conséquente, la multiplication des sources nous permettra également de mesurer l'influence du biais entre domaines sur la performance des modèles et d'expérimenter différentes stratégies visant à prendre en compte ces biais. Ces sources ont des origines diverses. Il peut s'agir :

- de banques de données acquises par le Cerema, avant la thèse et sur lesquelles de précédents travaux ont déjà été menés [32, 33, 43] (tunnel de Rive-de-Gier) ;
- de jeux de données issus de travaux scientifiques (base CODEBRIM [40]), mis à disposition par leurs auteurs et ayant été réannotés par nos soins afin de les rendre conformes à nos besoins (*i.e.* vérité terrain définie au pixel près, correction des quelques erreurs d'annotation) ;
- de nouvelles acquisitions réalisées durant la thèse (aux entrées d'un tunnel piéton et sur un bâtiment universitaire).

2.1.2 Tunnel routier de Rive-de-Gier

Le tunnel de Rive-de-Gier (Loire) est un tunnel en béton mesurant environ 80 mètres de long. Le tunnel présente peu d'équipements, ces derniers se résumant à des passe-câbles, des repères catadioptriques ainsi qu'à quelques panneaux et panonceaux.

Quatre types d'anomalies sont visibles, à savoir des fers apparents, des zones humides, de fines fissures et des nids de cailloux.

Les fers apparents ressortent tout particulièrement au sein des images. Là où les parties saines du béton ont un aspect lisse, dépourvu d'aspérités, la perte de matière entourant les armatures métalliques est granuleuse et présente une teinte nettement plus claire que les zones saines. Ce sont donc des objets facilement repérables dans ce tunnel.

Les zones humides, en revanche, posent davantage de difficulté. En effet, si certaines d'entre elles sont aisément identifiables en raison de la présence d'eau liquide en surface au moment de l'acquisition, l'identification d'autres parties du revêtement visuellement proches est plus délicate. Dans ces dernières, un dépôt de matière est reconnaissable mais il est difficile d'en connaître l'origine. La seule information apportée par l'image n'est pas suffisante pour déterminer leur nature, y compris pour un expert.

Les nids de cailloux sont à peine discernables. Ils revêtent un aspect circulaire dont la texture est granuleuse. La difficulté à les repérer tient dans leur petitesse, qui permet mal de les distinguer de pertes ou de dépôts de matières.

Au niveau des conditions de captation, un système d'éclairage composé de projecteurs et fixé au véhicule d'acquisition a été nécessaire (voir figure 2.1), l'éclairement apporté par l'équipement du tunnel étant insuffisant et naturellement dirigé vers la route. Toutefois, les images sont globalement sombres et sous-exposées sur leurs bords (on parle de « vignettage »). Les images sont prises en séquence, c'est-à-dire que des photographies sont réalisées à intervalle régulier lorsque le véhicule est en mouvement, à raison d'une captation tous les 50 cm dans le sens d'avancement. En conséquence, deux images adjacentes vont présenter un fort recouvrement, de l'ordre de 90% dans notre cas. Un même objet va donc apparaître plusieurs fois dans nos images avec un angle et un éclairage variant légèrement. Comme le champ de vision des caméras ne permet pas de visualiser tout le revêtement, plusieurs passages sont nécessaires pour recueillir une représentation intégrale de l'ouvrage. Pour chacun de ces passages, chaque caméra est orientée selon un angle différent. Les prises de vue sont réalisées à une distance comprise entre 4 et 5 mètres de la paroi et couvrent environ 5 m^2 de surface. Ainsi, il y a peu de variabilité d'échelle liée à l'acquisition. La figure 2.2 donne un exemple d'image issu de ce tunnel.

FIGURE 2.1 – Système d'acquisition utilisé dans le tunnel de Rive-de-Gier et monté sur un véhicule. Ce système comprend 4 caméras et 10 projecteurs. Un groupe électrogène assure l'alimentation de l'ensemble des équipements.

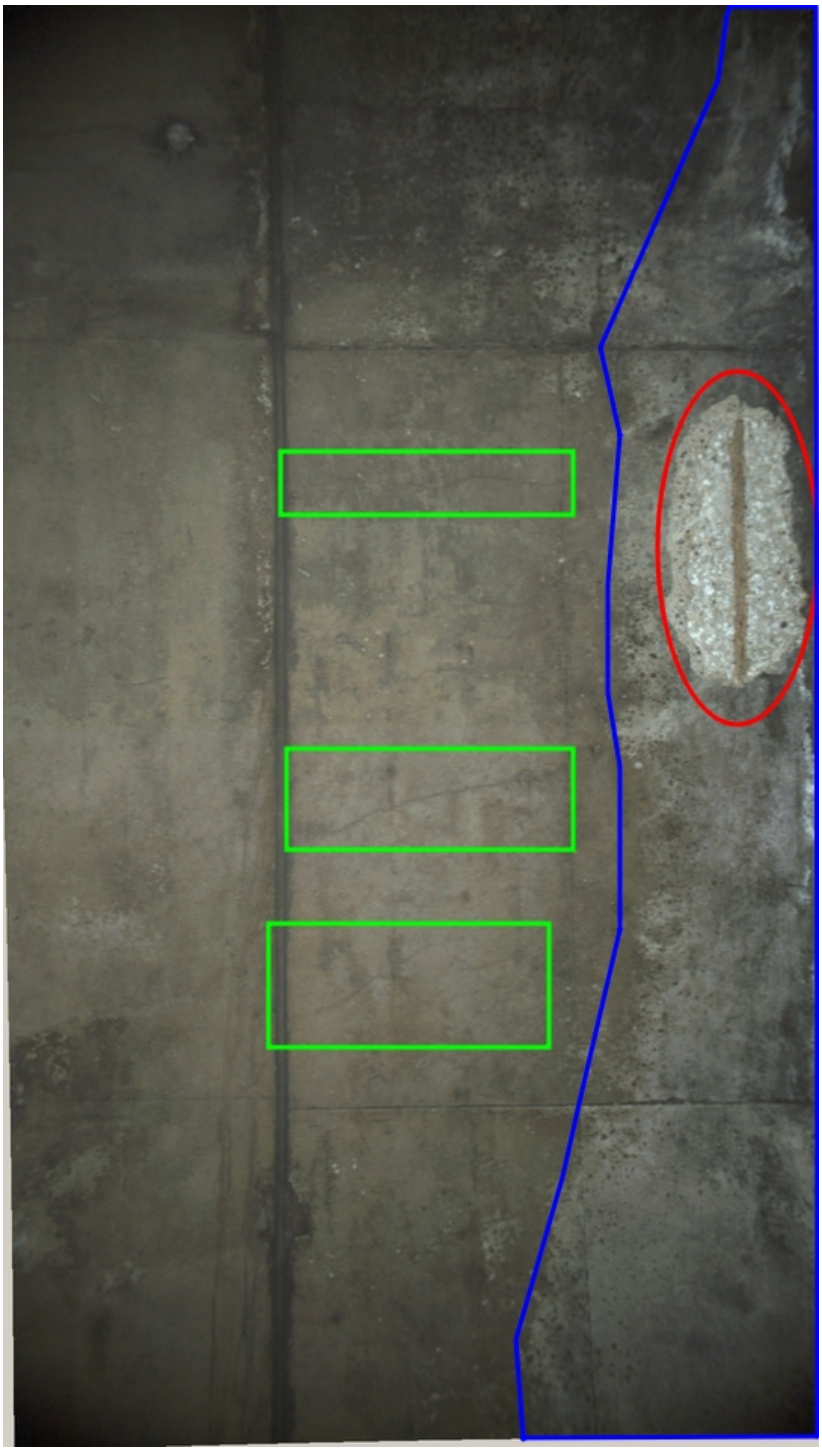


FIGURE 2.2 – Exemple d'image acquise dans le tunnel de Rive-de-Gier, au niveau de la clef de voûte de ce tunnel. Dans cette image, un fer apparent, entouré en rouge, est présent en haut à droite de l'image, quelques fissures fines, encadrées en vert, sont visibles au centre tandis qu'un dépôt de matière grisâtre, entouré en bleu et pouvant attester d'une zone humide traverse l'image de haut en bas dans la partie droite.

2.1.3 Base CODEBRIM

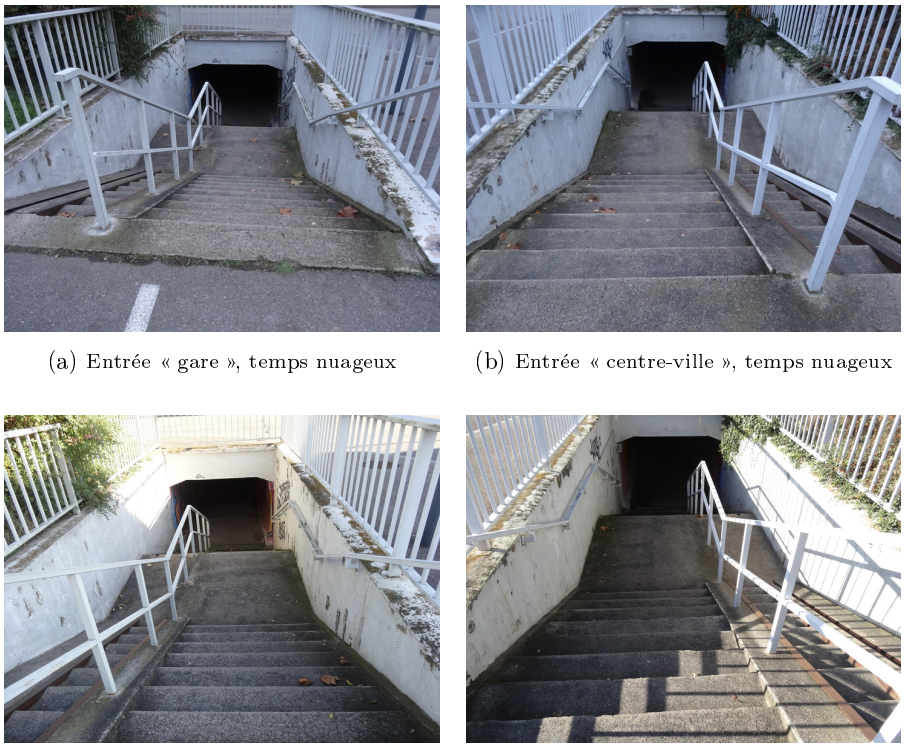
La base CODEBRIM (*COncrete DEfect BRidge IMage dataset*) [40] regroupe des photographies en haute résolution de trente ponts présentant des anomalies. Parmi toutes les sources de données que nous avons utilisées pour l'imagerie visible, CODEBRIM est sans doute la base qui présente le plus de variabilité, aussi bien en termes de luminosité, de distance d'observation par rapport à la paroi mais aussi de variabilité intrinsèque des anomalies (sévérité, forme, etc.).

Cette base répertorie différents types d'anomalies (fissures, efflorescences, épaufrures, fers apparents et zones corrodées). Néanmoins, comme les autres sources de données ne présentent, pour l'essentiel, que des fers apparents, nous avons restreint cette base aux images dans lesquelles figure ce type d'anomalie. Ainsi, des 1590 images qui composent initialement CODEBRIM, nous en retenons 366. Quelques exemples de cette base sont présentés en figure 2.3.

FIGURE 2.3 – Exemples extraits de la base CODEBRIM et présentant des fers apparents, à différentes échelles et sous différents angles de prise de vue.

2.1.4 Entrées de tunnel piéton

Le tunnel piéton est situé à Benfeld (Bas-Rhin) et est constitué de deux zones distinctes présentant des anomalies, à savoir ses deux entrées. Comme il relie le quartier gare avec le reste de la ville, ces deux entrées seront respectivement appelées « gare » et « centre-ville ». Cet ouvrage a été acquis dans deux conditions météorologiques différentes : par temps nuageux et par temps ensoleillé (voir figure 2.4).



(a) Entrée « gare », temps nuageux

(b) Entrée « centre-ville », temps nuageux

(c) Entrée « gare », temps ensoleillé

(d) Entrée « centre-ville », temps ensoleillé

FIGURE 2.4 – Les deux entrées du tunnel piéton situé à Benfeld, dans deux conditions météorologiques différentes. Des fers apparents sont présents sur les parois le long de la descente.

Si les images prises par temps nuageux présentent une illumination homogène, celles prises sous un grand soleil témoignent d'une plus grande variété d'aspects liée aux conditions d'éclairages. Comme le tunnel est orienté selon un axe est/ouest et que l'acquisition a eu lieu sous un soleil rasant de fin d'après-midi, un côté se retrouve intégralement à l'ombre tandis que l'autre baigne dans la lumière, et ce pour chaque entrée. Du côté ombragé, la paroi est généralement assez sombre et, du fait que l'objectif est tourné vers le soleil, quelques aberrations optiques sont présentes (facteur de *flare*). Sur le mur

éclairé, on peut observer les ombres projetées du pan opposé ainsi que celles des garde-fous le surplombant. Ces ombres se superposent parfois avec les anomalies recherchées. Enfin, un problème de dynamique de la photographie est notable pour les images comportant simultanément des zones ombragées et éclairées. Ainsi, la dynamique ne permet pas d'apprécier tous les éléments de l'image, le contraste étant nécessairement insuffisant dans l'une des deux zones.

En termes d'équipements, on retrouve des garde-corps, un rail pour le passage des vélos le long des descentes ainsi que des passe-câbles et des grilles de caniveaux au niveau des jonctions entre la descente et le tunnel en lui-même. Parmi les éléments distracteurs figurent un grand nombre de tags et d'écaillements de peinture. On peut aussi relever la présence de quelques végétaux (lierre) qui « tombent » sur les parois et viennent par endroit occulter des fers apparents.

La figure 2.5 présente quelques exemples provenant de ce tunnel.

(a) (b)

(c) (d)

FIGURE 2.5 – Exemples issus du tunnel piéton. Dans l'image (a), des tags recouvrent la paroi tandis qu'un facteur de *flare* est visible sous forme de tache verdâtre dans le fer apparent en haut de l'image. Dans (b), des écaillements de peintures, dont l'aspect les rapproche d'une perte de matière, sont observables. Les exemples (c) et (d) représentent pratiquement la même zone et illustrent le problème de la dynamique des images.

FIGURE 2.6 – Exemples d'images du tunnel piéton représentant la même zone avec différents points de vue. Sur chaque image, les deux mêmes fers apparents sont présents.

Enfin, il existe, dans cette base, une grande variabilité de points de vue entre la paroi et l'objectif. Ainsi, une même zone est généralement photographiée en utilisant différents angles et distances entre le revêtement et le capteur. Cette diversité a tout particulièrement été recherchée lorsque la zone acquise présentait une anomalie. La figure 2.6 illustre cet aspect à travers un exemple.

2.1.5 Bâtiment universitaire

Les images du bâtiment universitaire ont été acquises avec deux capteurs différents : un *smartphone* Apple et une tablette Samsung. Par la suite, nous appellerons A les images provenant du premier capteur et S celles du second.

Ce bâtiment présente une centaine de fers apparents, principalement situés au niveau du mur du $4^{\text{ème}}$ étage. Les photographies étant acquises depuis le sol, les images sont prises en contre-plongée et la distance entre le capteur et la paroi est de l'ordre de 10 mètres. On y trouve de nombreuses fenêtres, d'autres structures architecturales (piliers, etc.) ainsi que des lampadaires. Quelques arbres sont particulièrement occultants sur certaines images et masquent des anomalies. Bien que les conditions météorologiques lors de l'acquisition aient été proches pour les deux captations, les images de S sont plus sombres que celle de A. La figure 2.7 présente quelques images du bâtiment selon les deux capteurs.

FIGURE 2.7 – Exemples de prises de vue du bâtiment universitaire avec les deux capteurs. La colonne de gauche correspond aux images de S tandis que la colonne de droite reprend des exemples représentant approximativement la même zone mais provenant de A.

2.1.6 Jeux de données

2.1.6.1 Base pour la classification

Cette base est reprise de travaux antérieurs [33] et est constituée de deux jeux issus du tunnel routier de Rive-de-Gier. Ils sont constitués de sous-images de taille 101×101 pixels extraites d'images de résolution 1912×1081 et sont dédiés à la reconnaissance des quatre anomalies présentes dans ce tunnel : fissures, zones humides, fers apparents et nids de cailloux. Comme les jeux de données ne comportent que relativement peu d'exemples, il a été choisi de regrouper tous les exemples d'anomalies au sein d'une même classe. Chaque sous-image est donc annotée selon deux labels : revêtement sain ou anomalie. La composition de ces jeux est donnée dans le tableau 2.1.

	Apprentissage	Test
Sain	594	77
Anomalie	600	201

TABLEAU 2.1 – Composition (nombre d'exemples) des deux jeux de données

Comprenant 1472 exemples au total, cette base est donc plutôt réduite au regard du tunnel dont elle est extraite puisqu'en termes de superficie (*i.e.* nombre de pixels), elle correspond approximativement à 7 images acquises dans ce dernier (les séquences réalisées dans ce tunnel comptabilisent 1568 images). La figure 2.8 présente quelques exemples de cette base.

(a) Sain (b) Anomalies

FIGURE 2.8 – Exemples de sous-images de la base d'apprentissage ou de test. Les sous-images de la classe anomalie représentent de gauche à droite et haut en bas : un fer apparent, une zone humide calcifiée, une fissure et un nid de cailloux.

2.1.6.2 Base pour la segmentation sémantique

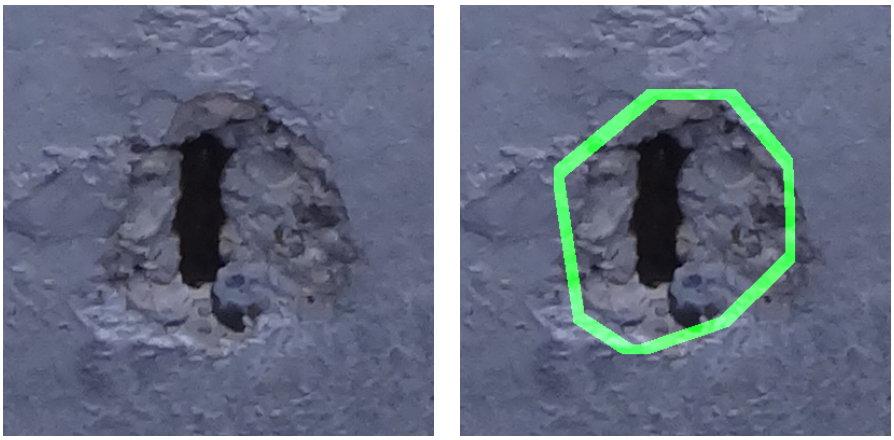
Pour toutes les sources de données présentées dans ce chapitre, nous avons annoté les fers apparents au pixel près. Plus précisément, nous considérons qu'un fer apparent est, visuellement parlant, défini comme une ou plusieurs armatures métalliques entourées d'une perte de matière, à la texture généralement granuleuse. Pour des raisons de facilité d'annotation, nous attribuons le même label à l'armature et à la perte de matière. Aussi, notons qu'une perte de matière seule n'est pas labellisée « fer apparent ». Ce choix d'annotation n'est pas anodin. Il peut induire une incohérence dans le processus d'apprentissage et d'évaluation si les données sont présentées de façon partielle. Par exemple, si une portion d'une zone étiquetée comme fer apparent est soumise à un modèle statistique alors que cette zone ne comporte pas d'armature métallique, le modèle est supposé répondre qu'il s'agit d'un revêtement sain. Il n'a, en effet, pas suffisamment de contexte pour affirmer le contraire. Or, cette zone a bien « fer apparent » pour label, d'où l'incohérence. S'il est difficile d'éviter de façon certaine la survenue d'une telle situation, il demeure nécessaire de veiller à en contenir le risque. La figure 2.9 illustre ce phénomène.

FIGURE 2.9 – Exemple de sous-image problématique (matérialisée en rouge). Elle a une résolution de 256×256 pixels et est intégralement annotée comme fer apparent mais l'armature métallique n'y figure pas.

Ce processus de labellisation a été réalisé grâce à un logiciel développé par

mes soins (*cf.* annexe A). Les annotations sont renseignées via des polygones dont l'utilisateur saisit les sommets de manière à entourer les anomalies. Pour réduire le temps d'annotation, le décalage entre deux images successives est calculé afin que les polygones précédemment créés par l'utilisateur dans une image puissent être repositionnés dans l'image suivante. Ce décalage est déterminé par appariement de points caractéristiques (descripteurs SIFT [44]). Ainsi, les anomalies communes aux deux images ne nécessitent qu'un léger ajustement. Cette fonctionnalité est particulièrement adaptée aux acquisitions réalisées en séquence comme celles du tunnel routier de Rive-de-Gier. Une fois tous les polygones fixés, ils sont discrétisés de sorte que chaque pixel des images les contenant se voit attribuer une étiquette de classe pour former la vérité terrain. Les polygones ne jouent alors plus aucun rôle dans la suite de la chaîne de traitement.

Pour tenir compte de l'imprécision des annotations réalisées à l'aide de polygones englobants, nous avons choisi de « neutraliser » la zone avoisinant ces derniers. Cette neutralisation consiste, lors de l'apprentissage, à assigner un coût nul à chaque pixel se trouvant dans le voisinage en question. Lors de l'évaluation, elle se matérialise par une mise à l'écart des pixels concernés lors du calcul des indicateurs de performance du modèle. Les zones neutralisées sont obtenues par dilatation morphologique des polygones discrétisés par un élément structurant carré dont le côté a été empiriquement fixé pour chaque site (20×20 pour le tunnel piéton et CODEBRIM ; 5×5 pour le tunnel routier et le bâtiment universitaire). La figure 2.10 présente le résultat de ce procédé sur un exemple.



(a) Exemple de fer apparent (b) Zone neutralisée (en vert)

FIGURE 2.10 – Illustration du procédé de neutralisation du voisinage des anomalies sur un fer apparent issu du tunnel piéton. Dans (b), la zone verte est neutralisée, son intérieur correspond à l'annotation de fer apparent tandis que l'extérieur est labellisé comme revêtement sain.

Les données de chacune des sources ont été subdivisées en trois jeux différents : apprentissage, validation et test. Pour mesurer le plus fidèlement possible les performances de nos modèles, il est préférable que les jeux extraits d'un même site mais de rôles différents soient disjoints, c'est-à-dire que deux images représentant une même zone physique n'appartiennent pas à des jeux différents. Il s'ensuit que les jeux peuvent uniquement être construits comme réunion d'ensembles d'images vérifiant transitivement cette condition.

Une fois les jeux définis, leur nombre détermine les rôles attribués. Lorsqu'un seul jeu est constitué, nous l'employons pour le test des modèles. Si deux jeux sont formés, l'un est destiné à l'apprentissage et l'autre au test. Un jeu de validation est ajouté aux jeux d'apprentissage et de test si un troisième jeu est composé. Pour chaque source, les jeux ainsi formés sont les suivants :

Tunnel routier de Rive-de-Gier Le tunnel de Rive-de-Gier a été séparé en deux sous-ensembles selon le sens longitudinal, c'est-à-dire que la démarcation peut être représentée par un plan transversal à l'axe de circulation. Comme il y a un recouvrement important entre deux images adjacentes, il a fallu exclure 140 images situées à l'extrémité commune aux deux sous-ensembles pour respecter la contrainte de disjonction des jeux. Puisque ce procédé se traduit par une perte de données, il a été choisi de ne créer que deux jeux (apprentissage et test). Comme l'écrasante majorité des images sont dépourvues de fers apparents, nous évaluons également nos modèles sur un jeu de test restreint aux images qui présentent ces anomalies. Ce jeu de test est qualifié par la suite de **restreint**.

CODEBRIM Pour cette base, le découpage choisi est le même que celui employé par les créateurs de la base [40]. On y retrouve donc trois sous-ensembles, correspondant aux trois jeux (apprentissage, validation et test).

Tunnel piéton Le tunnel piéton a été subdivisé en deux sous-ensembles. Un premier jeu, d'apprentissage, est constitué des prises de vue de l'entrée gare et un second, dédié au test, se compose des images de l'entrée côté centre-ville.

Bâtiment universitaire (S et A) Du fait du peu d'images recueillies sur cette structure, nous avons opté pour ne pas séparer les images en plusieurs sous-ensembles. Ces jeux sont, ainsi, exclusivement consacrés aux tests.

Le détail de la composition des jeux d'images est donné dans le tableau 2.2. On peut, en particulier, relever que les fers apparents occupent une faible proportion de l'ensemble des pixels pour tous les jeux à l'exception de ceux de CODEBRIM. L'histogramme de la figure 2.11, représente la distribution des tailles des fers apparents pour chaque site. Il en ressort que ces anomalies admettent des aires s'étendant sur plusieurs ordres de grandeur, y compris au sein d'un même site. Cette tendance est cependant moins marquée pour les deux captations du bâtiment universitaire ainsi que pour le tunnel de Rive-de-Gier.

Source	Taille des images	Apprentissage	Validation	Test
Rive-de-Gier	1912×1081	1010 **(9)**	–	418 **(9)**
Rive-de-Gier (R)	1912×1081	–	–	27 **(8)**
CODEBRIM	Variable	315 **(10)**	22 **(9)**	29 **(9)**
Tunnel piéton	4896×3672	134 **(9)**	–	78 **(9)**
Bâtiment univ. (S)	2048×1536	–	–	17 **(8)**
Bâtiment univ. (A)	4032×3024	–	–	56 **(9)**

(a) Nombre d'images au sein de chaque jeu. Les valeurs entre parenthèses renseignent le $\log 10$ du nombre de pixels présents dans le jeu correspondant.

Source	Apprentissage	Validation	Test
Tunnel de Rive-de-Gier	614	–	35
CODEBRIM	670	47	47
Tunnel piéton	409	–	100
Bâtiment universitaire (S)	–	–	197
Bâtiment universitaire (A)	–	–	454

(b) Nombre de fers apparents (exprimés en nombre de composantes 8-connexes dans la vérité terrain) pour chaque jeu.

Source	Apprentissage	Validation	Test
Tunnel de Rive-de-Gier	01.07	–	00.09
Tunnel de Rive-de-Gier (R)	–	–	01.46
CODEBRIM	19.81	22.57	24.71
Tunnel piéton	06.88	–	02.74
Bâtiment universitaire (S)	–	–	00.63
Bâtiment universitaire (A)	–	–	01.41

(c) Proportion de fers apparents (exprimés en pourcentage parmi les pixels présents) pour chaque jeu.

TABLEAU 2.2 – Composition des jeux de données photographiques utilisés pour la cartographie des fers apparents. La différence entre les deux jeux relatifs au bâtiment universitaire est expliquée en section 2.1.5. Pour le tunnel de Rive-de-Gier, le marqueur « (R) » fait référence au jeu restreint de ce même tunnel.

Outre cette variabilité de superficie, il convient de souligner que les fers apparents présentent, plus généralement, des différences d'aspect importantes d'un site à l'autre (teinte des armatures métalliques et des pertes de matière les entourant, orientations des armatures métalliques, présence de moisissure ou de mousse, conditions d'illumination, etc.). Le biais de domaine entre les différents sites est donc conséquent.

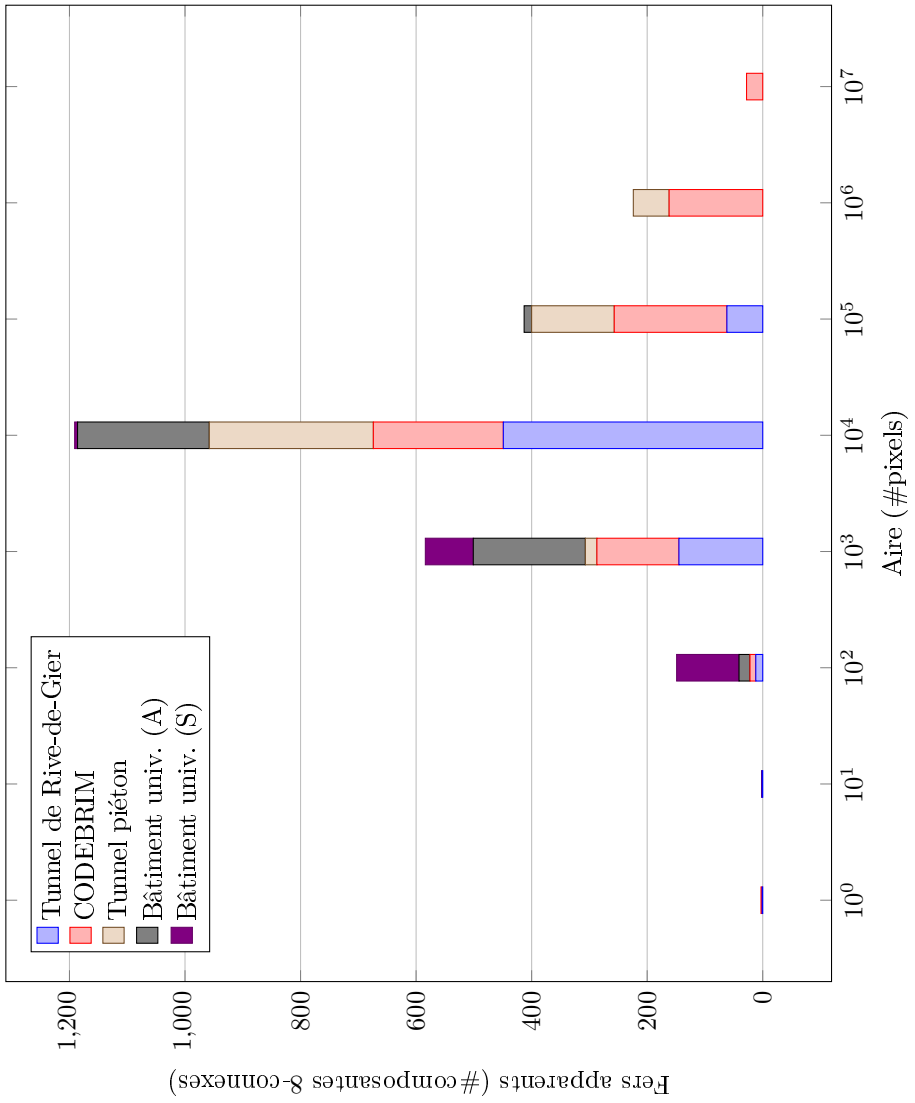


FIGURE 2.11 – Répartition des aires (en ordre de grandeur sur les pixels) des fers apparents (identifiés par les composantes 8-connexes de la vérité terrain) pour l'ensemble des sites d'acquisition.

2.2 Relevés laser LCMS

2.2.1 Généralités

Les données LCMS utilisées dans la thèse proviennent d'un seul site : le tunnel routier de la Grand Mare (Seine-Maritime). Il s'agit d'un tunnel à deux tubes d'une longueur de 1,5 km comportant de nombreux équipements : dispositifs d'aération, bornes d'incendie, issues de secours, etc. Les relevés de ce tunnel ont été recueillis dans le cadre d'une campagne d'acquisition menée par le Cerema et réalisée durant l'hiver 2018 [3]. Pour mener à bien cette entreprise, le Cerema a développé, avec l'aide du centre d'étude et de construction de prototypes (CECP) du Cerema-Rouen, un prototype d'acquisition, baptisé *MALT* (pour *Mobile Acquisitions with Lasers in Tunnels*), composé d'une camionnette sur lequel est fixé un transpalette servant de support à deux capteurs LCMS (*cf.* figure 2.12). Ce transpalette permet d'ajuster la distance et l'angle entre ces derniers et le revêtement du tunnel à acquérir. À l'instar du tunnel de Rive-de-Gier, l'acquisition se fait depuis un véhicule en mouvement et plusieurs passages de ce véhicule sont nécessaires pour obtenir une représentation intégrale de la paroi.

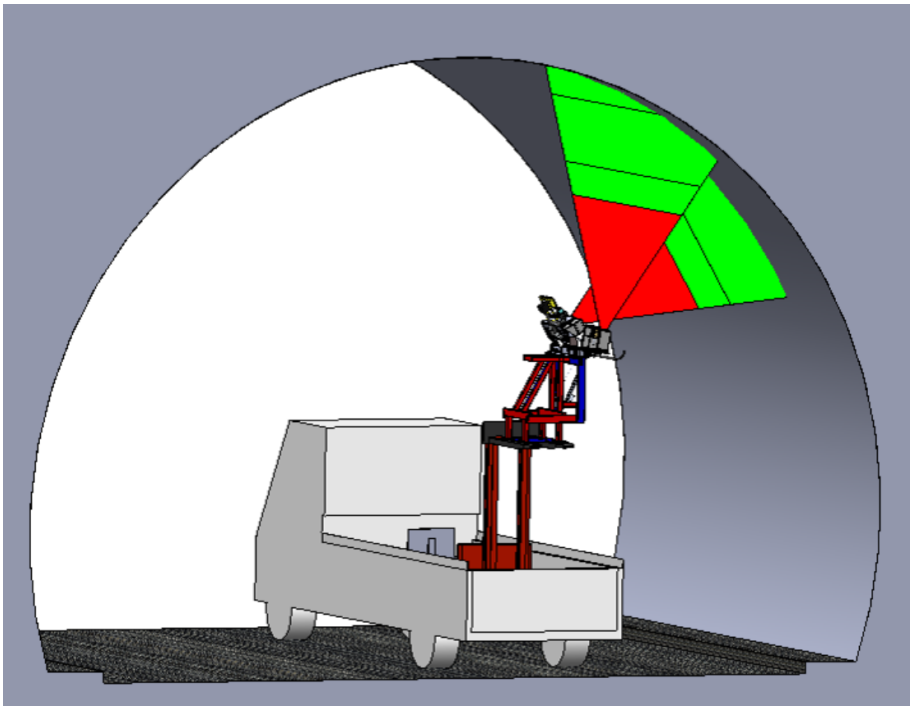
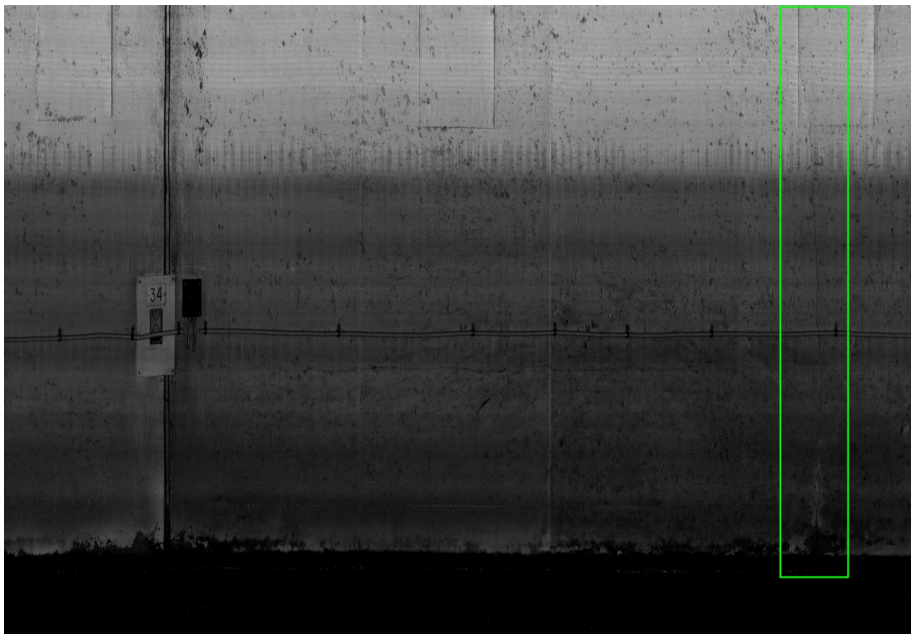


FIGURE 2.12 – Schéma représentant le prototype d'acquisition *MALT* (source : [3])

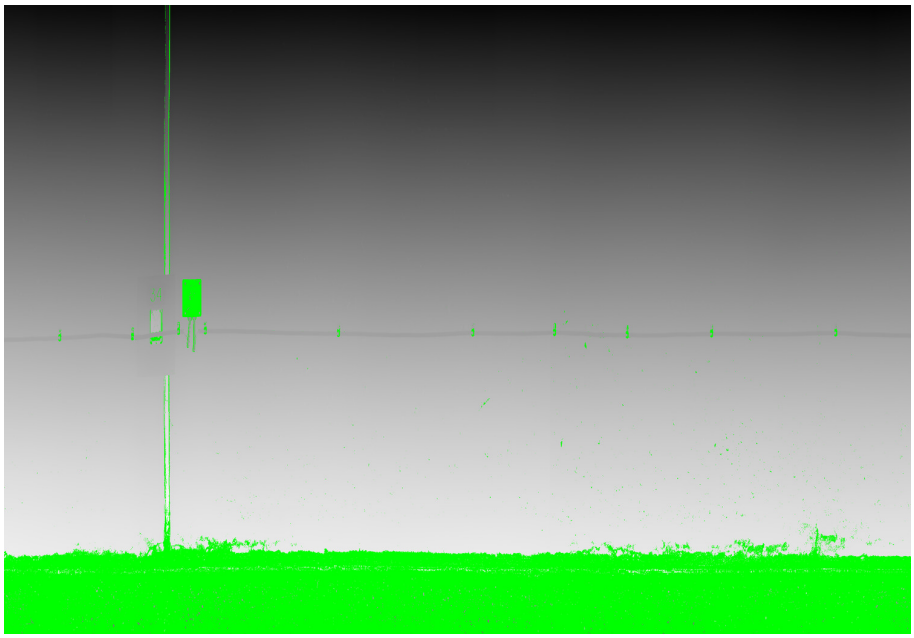
2.2.2 Fonctionnement du capteur LCMS

Pour réaliser la captation d'un revêtement, le capteur LCMS projette sur ce dernier une ligne laser dans le domaine infrarouge. Cette ligne est ensuite filmée par une caméra numérique. Les données produites par le capteur sont alors composées de deux canaux d'information synchronisés (directement par le capteur) : une première composante restitue l'intensité lumineuse de la ligne observée, et une seconde, calculée à partir des déformations de cette même ligne, en renseigne la profondeur. Cette information correspond à la distance entre le capteur et la surface et est exprimée en millimètres. Les lignes acquises sont alors juxtaposées pour produire des images. En termes de résolution, on dispose d'un point par millimètre au sein d'un même profil et on réalise l'acquisition d'un profil tous les deux millimètres. Cette résolution est compatible avec une vitesse de déplacement du véhicule de 8 km/h. Il est techniquement possible d'acquérir un profil tous les millimètres, mais cela implique de réduire la vitesse du véhicule à 4 km/h [3], ce qui représente une lourde contrainte opérationnelle. Lors de l'acquisition, il peut arriver que certains profils soient incomplets, en raison de problèmes de profondeur de champ (liés à des variations brusques de la forme du tunnel ou de la trajectoire du véhicule porteur) ou bien à cause de phénomènes optiques (diffraction, réflexion de la lumière) liés aux propriétés des objets rencontrés. Pour représenter ces données manquantes, une valeur nulle est renseignée en intensité tandis qu'en profondeur, la valeur « -10000 » est enregistrée. Par la suite, nous qualifions ces points de « hors de portée ». La figure 2.13 présente les deux composantes d'un même exemple de donnée LCMS.

Si les données fournies par le capteur LCMS sont fortement résolues, nous permettant d'espérer pouvoir localiser finement les anomalies, nous y constatons cependant un certain nombre d'artefacts. Premièrement, le capteur acquérant des profils à intervalles réguliers, on observe des déformations géométriques parfois conséquentes. Ces déformations sont généralement causées par une déviation de trajectoire du véhicule réalisant l'acquisition ou encore par des variations au niveau de la paroi. Deuxièmement, on peut noter la présence d'un bruit en tout point des images, possiblement causé par un défaut de calibration du capteur. Ce bruit, qui apparaît tant dans la carte d'intensité que dans la composante de profondeur, semble structuré mais n'est pas régulier pour autant. Il n'est donc pas possible de le supprimer par analyse fréquentielle sans perdre d'information utile. La figure 2.14 illustre ces deux types d'artefacts à travers deux exemples représentés par leur carte d'intensité. Il n'est pas établi que ces imperfections soient des facteurs de perturbation dans le cadre de notre application : on peut émettre l'hypothèse que les réseaux de neurones sont capables d'apprendre à en faire abstraction. Néanmoins, elles constituent une source supplémentaire de variabilité d'aspect dans les données et peuvent contribuer, si elles varient d'un site d'acquisition à un autre, à accroître le biais de domaine entre ces sites.



(a) Intensité



(b) Profondeur

FIGURE 2.13 – Composantes d'un exemple de donnée LCMS. Dans la composante de profondeur, les points hors de portée sont représentés en vert. Une fissure, encadrée en vert dans la composante d'intensité, traverse l'image de haut en bas sur la droite de celle-ci.

(a) Déformation géométrique (b) Bruit structuré

FIGURE 2.14 – Exemple d'artefacts dans les données LCMS. Pour les deux exemples, seule la composante d'intensité est représentée.

2.2.3 Traitement de la profondeur

La présence des points hors de portée au sein de la composante de profondeur rend l'utilisation de cette dernière non triviale pour les méthodes d'apprentissage. De nombreuses approches ont été proposées [45, 46] pour répondre à cette problématique. Ces dernières se répartissent en deux catégories : les approches guidées, qui attribuent une profondeur aux points hors de portée en exploitant une autre composante d'information, présumée exempte de points de cette nature, et les approches non guidées, qui réalisent cette même tâche mais sans employer d'information additionnelle. Comme, dans le cas des données LCMS, les points hors de portée affectent aussi bien la composante d'intensité que la composante de profondeur, nous ne disposons pas de composante dépourvue de ce type de point. Les approches que nous avons mises en œuvre sont ainsi exclusivement non guidées.

Par-delà la prise en compte de ces points particuliers, il convient de noter que la normalisation des données est une problématique centrale pour la carte de profondeur. En effet, cette dernière rend compte de la complexité de la surface du tunnel, surface sur laquelle sont présents de nombreux équipements. Or, ces différents éléments présentent des différences de taille importantes : là les équipements admettent une surface apparente de l'ordre du décimètre, l'ouverture des fissures se mesure en millimètres. Réussir à « filtrer » les éléments les plus grands de la carte de profondeur pourrait permettre d'améliorer les performances des modèles appris.

Nous avons exploré trois approches. À l'exception de la première approche, il a été décidé, pour des raisons diverses que nous détaillons par après, d'appliquer les traitements proposés à des sous-images carrées de taille 256×256 pavant l'image sans recouvrement. Les images entières ayant une résolution de 3000×2080 pixels, les bords droits et bas sont rognés de sorte que la largeur et la hauteur soient des multiples de 256. L'image résultante a alors une résolution de 2816×2048 pixels. Dans un souci d'homogénéité, les images issues de la première approche sont également tronquées selon le même procédé. Notons que

pour l'ensemble de ces approches, nous ramenons les valeurs de la composante qui en résulte de façon linéaire dans $[0; 1]$ avant de présenter les données à un réseau de neurones, le minimum ayant 0 pour image, le maximum 1. Cette normalisation est réalisée pour chaque sous-image.

Bien que consacrées aux relevés laser de chaussées, d'autres approches de normalisation de la profondeur ont été explorées dans notre champ applicatif. Nous présentons certaines de ces méthodes dans le chapitre 4.

Profondeur brute La première approche, naïve, consiste à conserver la valeur -10000 des points hors de portée pour la considérer au même titre que les autres valeurs. Par la suite, nous appellerons « profondeur brute » cette prise en compte de la profondeur.

Profondeur centrée Pour la seconde approche, on cherche à assigner une valeur « raisonnable » aux points hors de portée. Puisque la distance entre la paroi et le capteur peut varier (comme en atteste le dégradé dans la figure 2.13b), choisir une unique valeur pour toute l'image ne semble pas idéal. Nous procédons alors localement en recentrant chaque point hors de portée à la valeur moyenne des autres valeurs valides de la composante de profondeur au sein de chaque sous-image. Si la sous-image est intégralement composée de points hors de portée, nous ramenons l'intégralité des valeurs à 0. Nous parlerons de « profondeur centrée » pour désigner le résultat de ce traitement.

Profondeur ajustée Dans la troisième approche, on implémente la méthode proposée dans [3, 37], et dont l'idée est d'effectuer une approximation locale de la surface du tunnel, considérée comme réunion de sous-images, via des surfaces décrites par une fonction polynômiale de faible degré. On parle de « profondeur ajustée ». L'utilisation de sous-images permet ici de limiter le risque de mauvaise régression des surfaces. En effet, la surface du tunnel étant complexe, sa modélisation par une surface polynomiale de faible degré risque d'aboutir à une surface éloignée du revêtement à l'échelle d'une image entière, rendant le modèle de régression inadapté.

Formellement, cette approche vise à trouver une fonction polynômiale P telle que, pour tout pixel de coordonnées (x, y), et dont on note la profondeur $D(x, y)$,

$$P(x, y) = D(x, y) + \varepsilon(x, y) \tag{2.1}$$

où $\varepsilon(x, y)$ est une quantité dont on cherche à minimiser la valeur absolue. On peut, dès lors, remplacer chaque $D(x, y)$ par

$$\begin{cases} -\varepsilon(x, y) & \text{si } D(x, y) \neq -10000 \\ 0 & \text{sinon} \end{cases} \tag{2.2}$$

Cette quantité est par la suite appelée **résidu**. Avec ce procédé, on passe d'un référentiel lié au capteur, où la distance mesurée correspond à la distance entre

la paroi et le capteur, à un référentiel relatif à la paroi. Dans ce dernier référentiel, les valeurs de profondeur acquièrent une sémantique plus riche pour la problématique de détection des anomalies. Ainsi, une valeur positive indique un élément « en creux » quand une valeur négative en renseigne un élément saillant. De plus, la projection des points hors de portée sur la surface n'interfère pas avec la détection des anomalies. En effet, un point parfaitement situé à la surface de la paroi n'est probablement pas une anomalie induisant une variation de profondeur.

Pour estimer les paramètres de P, il est nécessaire de réaliser une régression robuste. En effet, en plus des points hors de portée, de nombreux équipements se situent sur le revêtement et pourraient fausser l'estimation. En posant $\bar{\alpha} = \{\alpha_0; \alpha_1; \alpha_2; \alpha_3; \alpha_4; \alpha_5\}$, les fonctions polynomiales employées dans le cadre de la thèse sont de la forme

$$P_{\bar{\alpha}}(x, y) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 y + \alpha_4 y^2 + \alpha_5 xy \tag{2.3}$$

et comportent donc six paramètres à déterminer. Puisque la régression aux moindres carrés est particulièrement sensible aux valeurs extrêmes, nous réalisons une estimation de ces paramètres à l'aide d'un M-estimateur [47, 48]. Pour ce faire, nous employons l'algorithme IRLS (*Iteratively Re-weighted Least Squares*) [49]. Cet algorithme construit itérativement deux suites. La première, $(\bar{\alpha}_n)$, représente les paramètres de la fonction polynomiale inférée. La seconde, notée (w_n), permet de mesurer l'adéquation des données par rapport aux fonctions polynomiales de la première suite. L'idée est d'utiliser (w_n) pour pondérer l'importance accordée aux données lors de la régression des paramètres. Ainsi, plus une valeur est loin de la dernière surface inférée, moins il contribuera au calcul de la suivante. L'estimation de la surface s'améliore alors de proche en proche. Les points proches de la surface prédites sont appelés ***inliers*** tandis que ceux qui en sont éloignés sont qualifiés d'***outliers***. Pour mesurer l'adéquation des données par rapport à la surface, nous utilisons une fonction de pondération, dérivée du M-estimateur de Hebert & Leahy, et définie par

$$\omega(r) = \frac{1}{1 + r^2} \tag{2.4}$$

La suite (w_n) est alors définie récursivement par

$$\begin{cases} w_0(x, y) & = 1 - \mathbf{1}_{D^{-1}(\{-10000\})}(x, y) \\ w_{n+1}(x, y) & = \omega\left(P_{\bar{\alpha}_n}(x, y) - D(x, y)\right) \end{cases} \tag{2.5}$$

Notons que, pour w_0, nous marquons explicitement les points hors de portée comme inadéquats à la surface. On ne souhaite, en effet, pas en tenir compte lors de la détermination des paramètres de cette dernière. La suite $(\bar{\alpha}_n)$ est définie en fonction de la suite (w_n) et vaut, pour tout $n \in \mathbb{N}$,

$$\bar{\alpha}_n = \operatorname{argmin}_{\bar{\alpha}} \sum_{x=1}^{256} \sum_{y=1}^{256} w_n(x, y) \left(P_{\bar{\alpha}}(x, y) - D(x, y)\right)^2 \tag{2.6}$$

La récursion s'arrête lorsque la suite $(\bar{\alpha}_n)$ a convergé. Dans le cas où cette convergence ne serait pas atteinte, notre implémentation arrête l'algorithme après 200 itérations et conserve la dernière surface inférée. Pour les sous-images composées exclusivement de points hors de portée, on pose $\alpha_i = 0$ pour tout $i \in [\![0; 5]\!]$. Dans cette situation, la valeur de P ne revêt alors aucune importance puisque les valeurs hors de portée sont ultérieurement ramenées à 0 selon l'expression 2.2.

Si l'algorithme utilisé pour la régression de surface permet une relative robustesse aux valeurs aberrantes (*i.e.* les points hors de portée et les éléments distants de la paroi), ces dernières se traduisent par des résidus, aussi bien positifs que négatifs, dont l'amplitude excède largement la granularité de la paroi. C'est, en particulier, le cas pour les joints et les équipements qui peuvent représenter un résidu de plusieurs centimètres. Conserver ces éléments ne présentant pas un grand intérêt pour la reconnaissance des fissures, nous appliquons un double seuillage sur les résidus pour les écrêter. Concrètement, on tronque toutes les valeurs dont la valeur absolue dans le nouveau référentiel excède 10 millimètres. Ces valeurs sont donc ramenées à $\pm$ 10 selon leur signe. Le résultat de ce traitement appliqué à l'exemple de la figure 2.13 est donné en figure 2.15.

En plus de l'information de profondeur ajustée, l'algorithme de régression indique, à travers la suite (w_n) et pour chaque point, l'importance accordée à ce dernier lors de l'inférence de la surface. On parle de **carte des *outliers***. La figure 2.16 présente une telle carte pour l'image prise pour exemple. On peut constater que la fissure, située à droite de l'image, est visible, de même que l'ensemble des équipements et des joints. De plus, le bruit structuré est souvent considéré comme *outliers*. Notons que les zones proches de certaines jointures sont également considérées comme telles, ceci en raison de la complexité de la surface du tunnel. L'utilisation de cette carte au sein de nos modèles de cartographie peut ainsi représenter un intérêt, les *inliers* étant majoritairement composés de revêtements sains.

2.2.4 Jeux de données

Pour constituer l'ensemble de nos jeux de données, on dispose de 230 images de résolution 3000×2080 pixels représentant une séquence d'acquisition d'un piédroit d'environ 1,5 km. Cette séquence est marquée par l'absence d'équipements volumineux (dispositifs d'aération, rampes d'éclairage, etc.) et les seules anomalies qui y figurent sont des fissures, qui ont alors été annotées au pixel près.

Annotation des fissures Les annotations ont été obtenues de façon semi-automatique : dans une interface dédiée et développée spécifiquement pour la thèse (*cf.* annexe A), un opérateur désigne manuellement une extrémité $P_0 \in \mathbb{R}^2$ d'une fissure puis, en sélectionnant une seconde extrémité $P_1 \in \mathbb{R}^2$ de cette fissure, un algorithme [50, 51] utilisant le *fast-marching* [52] en effectue un relevé.

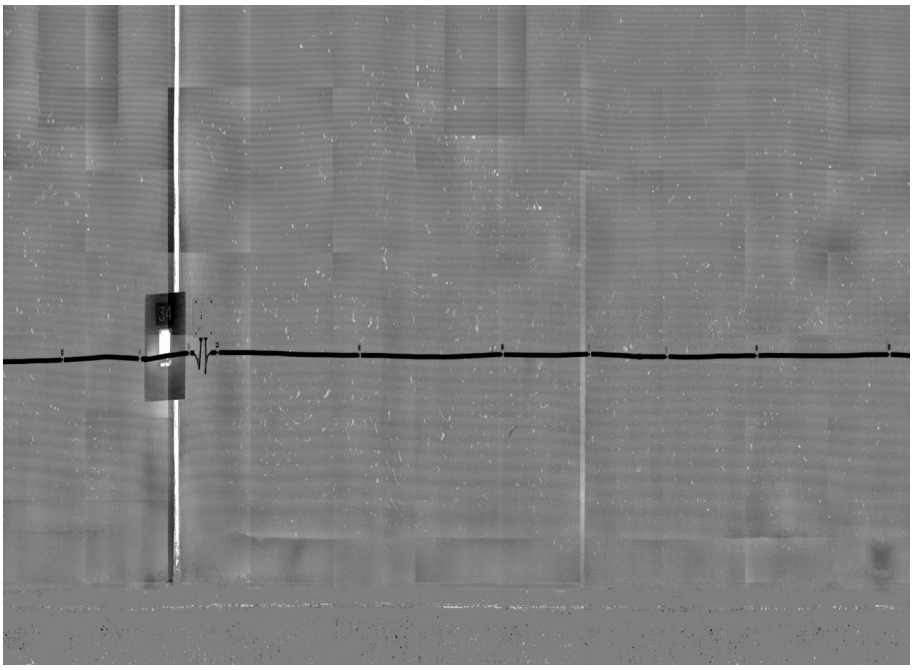


FIGURE 2.15 – Profondeur ajustée, avec troncature.

Cet algorithme repose sur le principe que les fissures sont généralement plus sombres que les revêtements sains. En notant $I\colon \mathbb{R}^2 \to \mathbb{R}$ l'application qui associe à chaque point l'intensité de l'image à annoter et

$$\Gamma_{M\to N} = \{\gamma\colon [0;1] \to \mathbb{R}^2 \mid \gamma(0) = M, \gamma(1) = N\} \tag{2.7}$$

l'ensemble des courbes continues reliant M à N, il s'agit de déterminer la courbe $\gamma \in \Gamma_{P_0\to P_1}$ minimisant la quantité $E(\gamma)$, définie par

$$E(\gamma) = \alpha \int_0^1 \|\gamma'(s)\| \mathrm{d}s + \int_0^1 I(\gamma(s)) \mathrm{d}s \tag{2.8}$$

où α est un hyper-paramètre contrôlant l'importance relative accordée à la longueur du chemin par rapport à la luminosité de ce dernier. Pour déterminer γ, on procède en deux temps.

1. Dans une première phase, on calcule, à partir du point P_0 sélectionné par l'utilisateur et pour tout point $P \neq P_0$ de l'image, la valeur

   $$\min_{\gamma\in\Gamma_{P_0\to P}} E(\gamma) \tag{2.9}$$

   Ce calcul est réalisé par propagation, depuis le point P_0, d'un front représentant, pour chaque point, le coût du chemin de plus petit coût reliant P_0 à ce point.

FIGURE 2.16 – Carte présentant les *outliers* issue de la régression robuste. Plus un pixel est sombre, moins il est pris en compte lors de la régression de la surface, et inversement.

2. Dans une seconde phase, on extrait le chemin en partant du point N par descente de gradient le long de la surface obtenue à l'étape 1.

Malgré l'emploi de méthodes semi-automatiques simplifiant le travail d'annotation, il convient de noter que l'obtention d'une vérité terrain de qualité reste une difficulté majeure. En effet, la largeur d'une fissure n'est pas constante le long de son tracé. Or, la méthode d'annotation produit des relevés de largeur fixe. De façon analogue à la base pour la segmentation sémantique, décrite dans la section 2.1.6.2, nous avons choisi de neutraliser la zone entourant les annotations. L'élément structurant utilisé pour la dilatation morphologique servant à la définition de cette zone est un carré de taille 5×5.

Constitution des jeux L'ensemble des images en pleine résolution est réparti en trois jeux de données deux à deux disjoints : un premier jeu dédié à l'apprentissage, un second consacré à la validation et un dernier réservé pour l'évaluation de la performance des modèles. Les méthodes de traitement de la profondeur que nous proposons imposant de travailler sur des sous-images de taille 256×256 pixels, on applique ce même découpage sur l'ensemble des images. Notons que l'extraction des sous-images n'est réalisée qu'après cette étape de répartition, si bien que deux sous-images issues d'une même image

ne peuvent donc pas se retrouver dans des jeux différents. Le tableau 2.3 présente le détail de cette répartition. Il en ressort que les fissures sont très peu représentées, aussi bien à l'échelle des sous-images que des pixels. Par ailleurs, la proportion d'exemples consacrés à chaque jeu (de l'ordre de 13% pour l'apprentissage, 13% pour la validation, et 74% pour le test) est plutôt inhabituelle dans le cadre de l'apprentissage machine, où la majeure partie des données est généralement dédiée à l'apprentissage. Ce choix est avant tout motivé par des contraintes matérielles. En effet, de nombreuses configurations étaient à évaluer pour les données LCMS. Il fallait donc que la durée d'exécution de chaque configuration testée demeure raisonnable, de l'ordre de quelques jours tout au plus. Pour garantir ce délai, il était indispensable que les jeux d'apprentissage et d'évaluation, sollicités à chaque époque de l'apprentissage, tiennent en mémoire vive. Le jeu de test, employé à une unique occasion après l'apprentissage, pouvait être chargé directement depuis la mémoire de masse. Ces éléments nous ont conduits à réduire la taille des jeux d'apprentissage et d'évaluation au profit du jeu de test.

	#Images	#Sous-images	% fissures (#s.i.)	% fissures (# pixels)
Apprentissage	30	2640 **(8)**	5.57	0.02
Validation	30	2640 **(8)**	7.80	0.03
Test	170	14960 **(9)**	6.30	0.02

TABLEAU 2.3 – Composition des jeux de données LCMS utilisés pour la cartographie des fissures. Les valeurs entre parenthèses renseignent le $\log 10$ du nombre de pixels présents dans le jeu correspondant. La troisième colonne indique la proportion de sous-images annotées comme fissures. La quatrième colonne renseigne cette même information, mais à l'échelle des pixels.

Conclusion du chapitre

À partir de deux types d'images, nous avons constitué plusieurs jeux de données permettant l'apprentissage, la validation et l'évaluation de modèles statistiques.

Images photographiques Deux bases de données ont été utilisées.

- La première, reprise de travaux antérieurs, est dédiée à la classification d'anomalies. Quatre types d'anomalies sont considérées (fissures, fers apparents, zones humides et nids de cailloux) et regroupées au sein d'une unique classe. Cette base a été construite de sorte que les classes soient équilibrées.
- La seconde, constituée par nos soins, répertorie les fers apparents au sein de cinq sources de données (deux portent sur des tunnels, deux autres sur un même bâtiment et une dernière sur des ponts). Ces fers apparents sont relevés au pixel près, permettant l'apprentissage de modèles de segmentation sémantique. Dans cette base, les fers apparents sont sous-représentés par rapport aux revêtements sains dans les images. De plus, on relève une grande variabilité d'aspect des fers apparents, aussi bien à l'intérieur de chacune des sources (variabilité intra-domaine) qu'entre elles (biais de domaine).

Données LCMS Une base a été constituée pour les relevés LCMS. Composée des prises de vue d'un des piédroits d'un tube du tunnel routier de la Grand Mare, elle est dédiée à la reconnaissance de fissures, qui sont annotées au pixel près. Les fissures étant fines, les pixels les constituant ne représentent qu'une faible proportion de l'ensemble des pixels. Cette base est aussi bien dédiée à la classification qu'à la segmentation sémantique.

Chapitre 3

Apprentissage profond et méthodologie d'évaluation

Dans ce chapitre, nous présentons les grands principes de l'apprentissage profond ainsi que la méthodologie d'évaluation des modèles statistiques. Loin d'être un catalogue exhaustif, ce chapitre a vocation à expliquer les techniques mises en œuvre dans le cadre de cette thèse ainsi qu'à motiver nos choix.

Sommaire

3.1 Apprentissage profond

Les réseaux de neurones constituent un modèle théorique et statistique assez ancien [53, 54]. Portée par un engouement important à ses origines, la recherche sur les réseaux de neurones voit ses financements se tarir à la suite de la parution de Minsky *et al.* [55] mettant au jour plusieurs limitations de ces derniers qui semblaient, à l'époque, difficilement surmontables. Bien que ces difficultés furent résolues dans le courant des années 80, les performances des réseaux de neurones demeuraient en deçà de celles des autres modèles. Les réseaux de neurones n'ont connu un formidable regain d'intérêt au sein de la communauté scientifique qu'à partir du début des années 2010 avec la victoire du réseau AlexNet au concours de reconnaissance d'images ILSVRC (*ImageNet Large Scale Visual Recognition Challenge* [56]), supplantant ainsi les autres méthodes, dont la majeure partie était non neuronale.

3.1.1 Réseau de neurones

3.1.1.1 Neurone formel

La brique élémentaire d'un réseau de neurones est le neurone formel. Le neurone formel est un modèle statistique minimaliste qui prend un certain nombre de réels en entrée $x_1, \ldots, x_n$ et qui en réalise une somme pondérée avant d'appliquer une fonction dite **d'activation** au résultat obtenu. Il existe de nombreuses fonctions d'activations (tangente hyperbolique/sigmoïde, ReLU [57], etc.), chacune ayant ses spécificités. Le point commun entre toutes ces fonctions tient dans leur non-linéarité. Les paramètres de ce modèle sont les poids appliqués pour la pondération ainsi qu'un paramètre additionnel, noté w_0 et appelé **biais**. La figure 3.1 donne une représentation d'un neurone formel.

En interconnectant plusieurs neurones formels entre eux, on obtient un modèle plus général qu'on appelle **réseau de neurones**. Le plus souvent, on arrange ces neurones par couches (*cf.* figure 3.2). On parle alors de **Perceptron multicouches** et les couches précédant la couche de sortie sont qualifiées de « cachées ».

Plus formellement, il est possible de voir le passage d'une couche de neurones à la suivante comme une application affine, dont on applique au résultat la

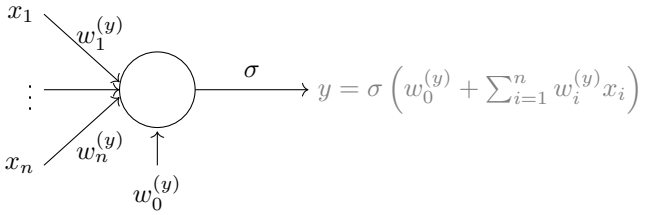


FIGURE 3.1 – Neurone formel. La fonction d'activation est notée σ et $w_0^{(y)}$ représente le biais du neurone.

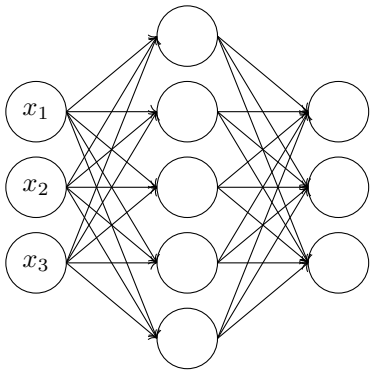


FIGURE 3.2 – Perceptron multicouches à 3 couches, présentant ainsi une couche cachée. Les biais et les fonctions d'activations sont omis afin d'éviter de surcharger le schéma.

fonction d'activation. Supposons que la première couche ait N_1 neurones (x_1, …, x_{N_1}) et que la seconde couche en ait N_2 (y_1, …, y_{N_2}). En posant

$$X = \begin{pmatrix} x_1 \\ \dots \\ x_{N_1} \end{pmatrix},\ Y = \begin{pmatrix} y_1 \\ \dots \\ y_{N_2} \end{pmatrix},\ W = \begin{pmatrix} w_1^{(y_1)} & \dots & w_{N_1}^{(y_1)} \\ & \vdots & \\ w_1^{(y_{N_2})} & \dots & w_{N_1}^{(y_{N_2})} \end{pmatrix},\ B = \begin{pmatrix} w_0^{(y_1)} \\ \dots \\ w_0^{(y_{N_2})} \end{pmatrix}$$

on a la relation

$$Y = \sigma(WX + B) \tag{3.1}$$

où σ est appliquée indépendamment sur chaque élément du vecteur.

Cette écriture matricielle fait transparaître l'intérêt de l'hypothèse de non-linéarité imposée pour les fonctions d'activation. En effet, une fonction d'activation linéaire rendrait tout perceptron multicouches égal à une application affine, limitant ainsi son expressivité. Réciproquement, il a été démontré par Cybenko [58] que les fonctions d'activations de type « sigmoïdes » permettent à un Perceptron composé de trois couches d'approcher avec une précision arbitraire toute fonction continue définie sur un compact de $\mathbb{R}^n$, pourvu que la couche cachée comporte suffisamment de neurones. Un résultat analogue a été démontré par Lu *et al.* [59] pour la fonction ReLU, à la différence que ce n'est plus le nombre de couches cachées qui est borné, mais le nombre de neurones par couches.

3.1.1.2 Réseaux de neurones appliqués aux images

Cette section est fortement inspirée de [60, 61]. La partie « analyse » en est une réécriture dans un cadre plus général et la partie « synthèse » est une de nos contributions.

Lorsque l'on souhaite mettre une image en entrée d'un réseau de neurones, l'approche naïve consiste à considérer un neurone pour chaque composante co-

lorimétrique de chaque pixel. Cependant, cette stratégie présente de nombreux écueils. Tout d'abord, le nombre de paramètres d'un tel réseau est colossal. En effet, un Perceptron à deux couches prenant comme entrée une image de $\mathcal{I}_{W,H,D}$ et ayant N neurones de sortie compterait $WHDN + N$ paramètres, ce qui est difficilement envisageable pour des images en haute résolution. De plus, chaque paramètre est spécifique à une zone spatiale de l'image et ne se généralise pas à d'autres. Or, cette situation n'est pas adaptée à notre problématique (ainsi qu'à l'essentiel des problématiques de traitements d'images), pour laquelle on souhaite qu'une anomalie soit reconnue de la même manière et ce, quelle que soit sa position dans l'image. À cette fin, il serait utile que le modèle repose, au moins en partie, sur des opérateurs commutant avec toute translation selon les dimensions spatiales de l'image. On parle d'équivariance par translation.

On souhaite ainsi construire une famille d'opérateurs vérifiant cette condition. Les architectures matérielles étant particulièrement adaptées au calcul matriciel, nous souhaitons également que ces opérateurs soient linéaires. Formellement, soit ϕ un tel opérateur et soit, pour tout $(x, y, z) \in \mathbb{Z}^3$,

$$\begin{aligned} t_{(m,n)}(x,y,z) \colon \mathcal{I}_{W,H,D} &\to \mathbb{R} \\ f &\mapsto f(x-m, y-n, z) \end{aligned} \tag{3.2}$$

l'application translatant du vecteur $(m, n) \in \mathbb{Z}^2$ une image, pour laquelle on convient que toute image est nulle en dehors de son domaine de définition. On veut que ϕ fasse commuter le diagramme suivant :

$$\begin{array}{ccc} \mathcal{I}_{W,H,D_1} & \xrightarrow{\phi} & \mathcal{I}_{W,H,D_2} \\ \downarrow{\scriptstyle t_{(m,n)}} & & \downarrow{\scriptstyle t_{(m,n)}} \\ \mathcal{I}_{W,H,D_1} & \xrightarrow{\phi} & \mathcal{I}_{W,H,D_2} \end{array} \tag{3.3}$$

où D_1 (respectivement D_2) est le nombre de canaux de l'image d'entrée (respectivement produite). On cherche ainsi à trouver des opérateurs linéaires et équivariants pour la translation quel que soit le nombre de canaux en sortie désiré.

On pose $\delta_{(m,n,k)}$ l'élément de $\mathcal{I}_{W,H,D_1}$ valant 1 aux coordonnées (m, n, k) et 0 partout ailleurs. L'ensemble de ces éléments forme donc une « base canonique » de $\mathcal{I}_{W,H,D_1}$. En effet, on a, pour tout $f \in \mathcal{I}_{W,H,D_1}$,

$$f = \sum_{k=1}^{D_1} \sum_{m=1}^{W} \sum_{n=1}^{H} f(m,n,k).\delta_{(m,n,k)} \tag{3.4}$$

On procède par analyse-synthèse.

Analyse Supposons qu'il existe un opérateur linéaire ϕ vérifiant les conditions du diagramme. On a :

$$\phi(f)(x,y,z) = \phi\left(\sum_{k=1}^{D_1}\sum_{m=1}^{W}\sum_{n=1}^{H} f(m,n,k).\delta_{(m,n,k)}\right)(x,y,z) \tag{3.5}$$

Par application de l'hypothèse de linéarité de ϕ, il s'ensuit

$$\phi(f)(x,y,z) = \left(\sum_{k=1}^{D_1}\sum_{m=1}^{W}\sum_{n=1}^{H} \phi\left(f(m,n,k).\delta_{(m,n,k)}\right)\right)(x,y,z) \tag{3.6}$$

$$= \left(\sum_{k=1}^{D_1}\sum_{m=1}^{W}\sum_{n=1}^{H} f(m,n,k).\left(\phi \circ \delta_{(m,n,k)}\right)\right)(x,y,z) \tag{3.7}$$

On peut alors expliciter l'opérateur de translation en décalant l'origine des éléments δ et utiliser l'hypothèse de commutativité du diagramme. On obtient

$$\phi(f)(x,y,z) = \left(\sum_{k=1}^{D_1}\sum_{m=1}^{W}\sum_{n=1}^{H} f(m,n,k).\left(\phi \circ t_{(m,n)} \circ \delta_{(0,0,k)}\right)\right)(x,y,z) \tag{3.8}$$

$$= \left(\sum_{k=1}^{D_1}\sum_{m=1}^{W}\sum_{n=1}^{H} f(m,n,k).\left(t_{(m,n)} \circ \phi \circ \delta_{(0,0,k)}\right)\right)(x,y,z) \tag{3.9}$$

En posant, pour tout $(x,y) \in \mathbb{Z}^2$, $h_{k,z}(x,y) = \left(\phi\left(\delta_{(0,0,k)}\right)\right)(x,y,z)$ et, pour tout $(m,n) \in \mathbb{Z}^2$, $f_k(m,n) = f(m,n,k)$, on peut réécrire cette expression de la façon suivante

$$\phi(f)(x,y,z) = \left(\sum_{k=1}^{D_1}\sum_{m=1}^{W}\sum_{n=1}^{H} f(m,n,k).\left(t_{(m,n)} \circ h_{k,z}\right)\right)(x,y) \tag{3.10}$$

$$= \sum_{k=1}^{D_1}\sum_{m=1}^{W}\sum_{n=1}^{H} f_k(m,n).h_{k,z}(x-m,y-n) \tag{3.11}$$

$$= \sum_{k=1}^{D_1}(f_k * h_{k,z})(x,y) \tag{3.12}$$

où $*$ désigne l'opérateur de convolution.

Synthèse Dès lors, si ϕ existe, son application consiste à réaliser des sommes de produits de convolution. En plus de démontrer l'existence d'une infinité d'opérateurs fonctionnels satisfaisant les critères recherchés, nous allons montrer que, quel que soit $h \in \mathcal{I}_{W,H,D_2}$, il existe un tel opérateur ϕ_h, vérifiant $\phi_h\left(\delta_{(0,0,k)}\right) = h$. On peut alors faire abstraction de l'opérateur linéaire sous-jacent et considérer directement les images h.

Soit Φ l'ensemble des opérateurs linéaires commutant avec les translations. On peut remarquer que les applications identité et identiquement nulle appartiennent à Φ. De plus, Φ est stable par somme et par composition (les preuves

étant immédiates, elles sont omises par concision). Il en découle que $(\Phi, +, \circ)$ est un anneau unitaire. On pose, pour tout $(x, y, z) \in \mathbb{Z}^3$

$$\begin{aligned} T_{(m,n,p)}(x,y,z) \colon \mathcal{I}_{W,H,D} &\to \mathbb{R} \\ f &\mapsto f(x-m, y-n, z-p) \end{aligned} \tag{3.13}$$

les opérateurs de translation tridimensionnelle. Remarquons que l'on a, pour tout $m, n, p, m', n' \in \mathbb{Z}$,

$$T_{(m,n,p)} \circ t_{(m',n')} = T_{(m+m',n+n',p)} = t_{(m',n')} \circ T_{(m,n,p)} \tag{3.14}$$

Ainsi, $T_{(m,n,p)} \in \Phi$ pour tout $m, n, p \in \mathbb{Z}$. On peut, dès lors, construire, pour tout $h \in \mathcal{I}_{W,H,D_2}$,

$$\phi_h = \sum_{p=1}^{D_2} \sum_{m=1}^{W} \sum_{n=1}^{H} h(m,n,p).T_{(-m,-n,-p+k)} \tag{3.15}$$

On a bien $\phi_h \in \Phi$ grâce à la structure d'anneau de ce dernier. De plus,

$$\phi_h\left(\delta_{(0,0,k)}\right) = \sum_{p=1}^{D_2} \sum_{m=1}^{W} \sum_{n=1}^{H} h(m,n,p).T_{(-m,-n,-p+k)}(\delta_{(0,0,k)}) \tag{3.16}$$

$$= \sum_{p=1}^{D_2} \sum_{m=1}^{W} \sum_{n=1}^{H} h(m,n,p).\delta_{(m,n,p)} \tag{3.17}$$

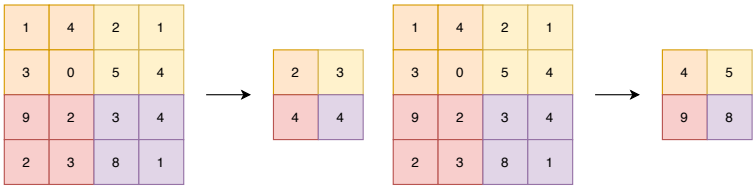
$$= h \tag{3.18}$$

Conclusion de la preuve et implications Les opérateurs commutant avec les translations bidimensionnelles sont exactement ceux consistant à effectuer une somme de produits de convolution sur l'image d'entrée. Ainsi, si l'on veut apprendre un tel opérateur, il suffit d'apprendre les valeurs constituant les opérandes de droite au sein des produits de convolution, dont il convient de fixer en amont les dimensions. L'ensemble de ces opérandes est alors appelé **noyau de convolution** et l'opération complète **couche de convolution**. Formellement, la définition d'une couche de convolution est donnée par l'expression 3.12, dans laquelle $h_{k,z}$ constitue le noyau de convolution et représente ainsi les paramètres de la couche de convolution.

L'image résultante de l'application d'une telle couche est appelée **carte de caractéristiques** ou encore **représentation**. De façon analogue aux Perceptrons multicouches, on ajoute un biais à l'issue du calcul de la convolution avant d'appliquer une fonction d'activation sur chaque élément de l'image produite. Le biais est un élément de $\mathcal{I}_{1,1,D_2}$. Il est ainsi ajouté à chaque pixel de la représentation.

3.1.1.3 Réseaux convolutifs

Bien qu'il soit techniquement possible de chaîner les couches de convolution, il est souvent préférable de réduire progressivement les dimensions spatiales



(a) *Average pooling* (b) *Max pooling*

FIGURE 3.3 – Illustration du fonctionnement des deux couches de *pooling* les plus fréquemment rencontrées, sur un même exemple.

des cartes de caractéristiques produites pour permettre au réseau de neurones de mieux synthétiser l'information. À cette fin, on introduit des couches qui vont sous-échantillonner les données issues de la couche qui les précède d'un facteur $r \in \mathbb{N}^*$. On parle de **couche de *pooling***. Pour ce faire, on introduit une fonction d'agrégation $A\colon [\![1;r]\!] \times [\![1;r]\!] \to \mathbb{R}$. Une couche de « A *pooling* », notée $\Pi_{A,r}$, est alors l'application définie par

$$\begin{aligned} \Pi_{A,r} \colon \mathcal{I}_{W,H,D} &\to \mathcal{I}_{W,H,D} \\ f &\mapsto \pi_{A,r}(f) \end{aligned} \tag{3.19}$$

avec

$$\begin{aligned} &\pi_{A,r}(f)\colon \mathbb{Z}^3 \to \mathbb{R} \\ &\quad (x,y,z) \mapsto A(f(x,y,z),\dots,f(x+r,y+r,z)) \end{aligned} \tag{3.20}$$

où les arguments de A correspondent au voisinage $r \times r$ de (x,y) dans l'image d'origine.

Généralement, la fonction A est soit la moyenne arithmétique (*average pooling*), soit la fonction maximum (*max pooling*). La figure 3.3 illustre le fonctionnement de ces deux exemples de couches de *pooling* avec $r = 2$ sur un même exemple. Les dimensions sont donc amenées de $4 \times 4 \times 1$ à $2 \times 2 \times 1$.

Conjugués avec les couches de convolution et les Perceptrons multicouches, ces différents éléments permettent de construire les réseaux convolutifs. De façon schématique, les réseaux convolutifs sont constitués d'une alternance de couches de convolution et de couches de *pooling*, ce schéma étant possiblement répété plusieurs fois, avant de se conclure par un Perceptron multicouches chargé d'inférer la prédiction (fig. 3.4). Dans ce contexte, chaque couche du Perceptron est appelée **couche complètement connectée**. Formellement, un réseau convolutif CNN peut être modélisé par :

$$\begin{aligned} &\mathrm{CNN}\colon \mathcal{I}_{W,H,D} \to \mathbb{R}^n \\ &\quad f \mapsto \left(\underbrace{\bigcirc_i (\sigma \circ P_i))}_{\text{Perceptron multicouches}} \right) \circ F \circ \left(\underbrace{\bigcirc_i (\Pi_{A,r} \circ C_i)}_{\text{Encodeur}} \right) \end{aligned} \tag{3.21}$$

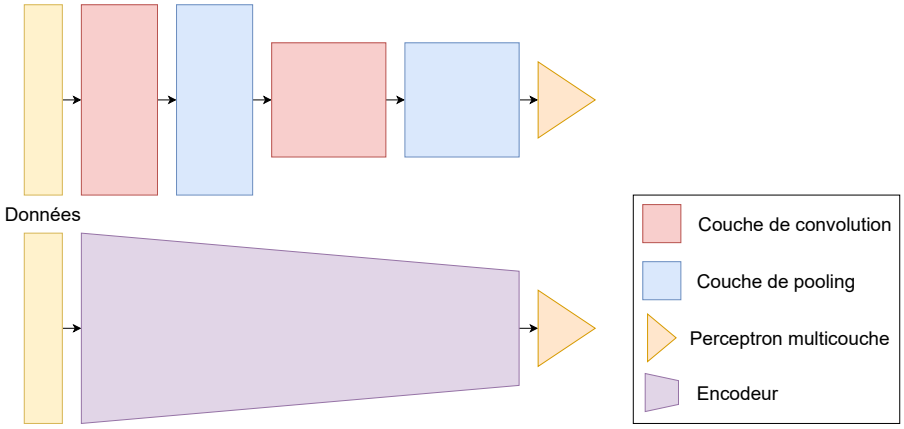


FIGURE 3.4 – Deux vues d'un même réseau convolutif. Vue détaillée en haut, synthétique en bas. Pour alléger le schéma, la couche de *flatten* n'est pas représentée.

où $F\colon \mathcal{I}_{W',H',D'} \to \mathbb{R}^{W'H'D'}$ est une fonction, appelée ***flatten***, aplatissant une image en un vecteur, où $\bigcirc_i e_i$ désigne la composition de N éléments e_i (i.e. $\bigcirc_i e_i = e_1 \circ \cdots \circ e_N$) et où chaque $\sigma \circ P_i$ est une couche complètement connectée au résultat duquel est appliqué, terme à terme, la fonction d'activation σ. La composante précédant les couches complètement connectées est appelée **encodeur** et est souvent représentée graphiquement par un trapèze isocèle, la grande base matérialisant les données d'entrée et la petite la sortie produite.

3.1.1.4 Apprentissage d'un réseau convolutif

L'apprentissage des paramètres d'un réseau convolutif suit les grandes lignes définies dans le chapitre introductif : on part d'une valeur d'initialisation et on modifie itérativement les paramètres en minimisant une fonction de coût par descente de gradient (algorithme de rétropropagation du gradient).

Initialisation Pour l'initialisation, une première approche consiste à initialiser les poids du réseau aléatoirement. À cette fin, plusieurs stratégies ont été proposées [62, 63]. Celle décrite par Glorot *et al.* [62] est l'option par défaut au sein des principales bibliothèques logicielles mettant en œuvre l'apprentissage profond (PyTorch, TensorFlow, etc.). Il s'agit, pour une couche complètement connectée à N neurones, d'assigner à chaque poids (y compris les biais), les réalisations de variables aléatoires identiquement distribuées selon une loi continue uniforme $\mathcal{U}(-\sqrt{k}, \sqrt{k})$, où $k = \frac{1}{N'}$ et N' est le nombre de neurones de la couche précédente. Pour une couche de convolution, le procédé est identique, à ceci près que k vaut $\frac{1}{HWD_1}$. Ici aussi, H, W et D_1 renvoient aux dimensions de la carte de caractéristiques précédant la couche de convolution.

Une seconde approche consiste à employer les poids résultant d'un apprentissage antérieur et à faire porter l'apprentissage sur tout ou partie des paramètres associés. On parle de ***fine tuning***. Cette pratique est particulièrement populaire en apprentissage profond puisque les communautés de chercheurs mettent régulièrement à disposition, pour les architectures de neurones les plus utilisées, les modèles appris sur les principaux jeux de données. Or, il a été démontré que les premières couches des réseaux de neurones ont tendance à extraire des motifs génériques (contours, points d'intérêt, etc.) et indépendants de la tâche considérée [64]. Dès qu'un modèle pré-appris et conforme à notre configuration expérimentale était disponible, nous avons opté pour cette seconde approche dans nos travaux.

Rétropropagation du gradient La rétropropagation du gradient [65] est un algorithme permettant de mettre à jour la valeur de chaque paramètre d'un réseau de neurones. Cette opération n'est pas complètement triviale puisque le réseau est construit comme une composition de nombreuses fonctions. Les paramètres ne sont donc pas tous en « contact » direct avec les valeurs d'entrée. Pour calculer le gradient pour chacun d'eux, on utilise le théorème de dérivation des fonctions composées.

Par exemple, en reprenant le modèle CNN défini par l'équation 3.21, le gradient selon un paramètre θ_c de la première couche de convolution C_1 et pour la fonction de coût $\mathcal{L}$ serait égal à

$$\frac{\partial}{\partial \theta_c} \mathcal{L} \circ \mathrm{CNN} = \frac{\partial \mathcal{L}}{\partial \mathrm{CNN}} \frac{\partial \mathrm{CNN}}{\partial C_1} \frac{\partial C_1}{\partial \theta_c} \tag{3.22}$$

Sur le plan calculatoire, ces opérations sont réalisées récursivement depuis la couche la plus proche de la sortie. Pour cette dernière couche, le gradient est calculé directement à partir de la fonction de coût appliquée au résultat et à la vérité terrain. Puis, le gradient de la couche précédente est déterminé à partir de celui de cette couche, et ainsi de suite.

Descente de gradient Une fois le gradient calculé, on peut l'utiliser pour mettre à jour les poids. Le premier algorithme à avoir été proposé pour cette tâche, et qui demeure encore employé de nos jours, est la descente du gradient stochastique (ou *SGD*, pour *Stochastic Gradient Descent*). Elle consiste à présenter les exemples annotés au modèle selon un ordre aléatoire et différent à chaque époque, avant d'utiliser l'équation 3.23 pour mettre à jour les poids (les notations employées sont définies dans le chapitre d'introduction, dont l'équation est reprise).

$$\theta^{(k+1)} = \theta^{(k)} - \mu \frac{1}{\#\mathcal{D}} \sum_{I \in \mathcal{D}} \nabla \mathcal{L} \left(f_{\theta^{(k)}}(I), \Gamma(I) \right) \tag{3.23}$$

Une alternative à SGD fréquemment rencontrée dans la littérature est Adam (*Adaptative momentum*) [66]. Là où SGD utilise un pas d'apprentissage μ fixe, Adam cherche à régulariser la descente de gradient en adaptant l'amplitude

de ce pas d'apprentissage et en appliquant une inertie (*i.e. momentum*) sur la direction du gradient considéré. Pour ce faire, Adam construit deux suites $(m_k)_{k\in\mathbb{N}}$ et $(v_k)_{k\in\mathbb{N}}$ à valeur dans $\mathbb{R}^{\#\theta}$. La première vise à estimer le moment d'ordre un du gradient (moyenne) pour chaque paramètre du modèle selon la fonction de coût considérée quand la seconde porte sur le moment d'ordre deux (variance non centrée) de cette même grandeur.

Initialisées toutes deux à $0_{\mathbb{R}^{\#\theta}}$, les suites sont construites récursivement par moyenne pondérée exponentielle. En posant $g_k = \nabla(\mathcal{L} \circ f_{\theta^{(k)}})$ le vecteur contenant la valeur du gradient de la fonction de coût $\mathcal{L}$, on a

$$m_{k+1} = \beta_1 m_k + (1-\beta_1) g_k \tag{3.24}$$

$$v_{k+1} = \beta_2 v_k + (1-\beta_2)(g_k \odot g_k) \tag{3.25}$$

où $\beta_1, \beta_2 \in [0;1[$ sont deux hyper-paramètres, et l'opérateur $\odot$ désigne la multiplication terme à terme. Pour faciliter la convergence de l'apprentissage, les auteurs d'Adam [66] recommandent de prendre $\beta_1 = 0,9$ et $\beta_2 = 0,999$. En raison de l'initialisation de ces deux suites, ces dernières ont tendance à être sous-évaluées, c'est-à-dire que chaque membre du vecteur est, en valeur absolue, plus proche de 0 que la valeur qui était à estimer. Pour corriger ce biais, on introduit deux autres suites $(\hat{m}_n)_{n\in\mathbb{N}}$ et $(\hat{v}_n)_{n\in\mathbb{N}}$ définie, pour tout $k \in \mathbb{N}$, $\hat{m}_k = \frac{m_k}{1-\beta_1^{\odot k}}$ et, symétriquement, $\hat{v}_k = \frac{v_k}{1-\beta_2^{\odot k}}$

L'équation de mise à jour des poids du modèle est alors modifiée de la façon suivante :

$$\theta^{(k+1)} = \theta^{(k)} - \alpha \frac{\hat{m}_{k+1}}{\sqrt{\hat{v}_{k+1}} + \varepsilon} \tag{3.26}$$

où $\alpha \in \mathbb{R}_+$ est un hyper-paramètre et $\varepsilon = 10^{-8}$.

En comparaison de *SGD*, la méthode Adam tend à converger plus rapidement (c'est d'ailleurs l'argument principal de ses auteurs). Cependant, certains travaux, comme ceux de Wilson *et al.* [67] ont mis en avant que les méthodes avec un pas adaptatif peuvent conduire à un modèle généralisant moins bien qu'un modèle appris avec *SGD*. À cette heure, il n'y a toutefois pas de consensus au sein de la communauté scientifique sur la question du choix de l'optimiseur.

Apprentissage par *batch* Les deux méthodes que nous avons vues effectuent une mise à jour des poids du modèle à chaque exemple. Il est souvent préférable, aussi bien pour des raisons de temps de calcul que de stabilité numérique, de décomposer le jeu d'apprentissage en sous-ensembles de taille restreinte, appelés ***batch***, et de considérer le gradient moyen sur l'intégralité du *batch*. Pour ce faire, il suffit, pour les deux méthodes de descente de gradient détaillées au paragraphe précédent, de remplacer $\nabla\mathcal{L}\left(f_{\theta^{(k)}}(I), \Gamma(I)\right)$ par $\frac{1}{\#B}\sum_{b\in B} \nabla\mathcal{L}\left(f_{\theta^{(k)}}(b), \Gamma(b)\right)$.

3.1.1.5 Couches spécifiques

En plus des couches de convolution et complètement connectées, il existe également d'autres types de couches, poursuivant chacune des objectifs spéci-

fiques.

Batch normalization Un exemple de couche couramment employée est la couche de ***batch normalization*** [68]. Généralement placées en enfilade des couches de convolution, ces couches de normalisation permettent d'améliorer la stabilité de l'apprentissage des réseaux de neurones. Cette stabilité est obtenue en renormalisant les cartes de caractéristiques par un centrage et une réduction de ces dernières. La particularité de cette méthode est que les moyennes et variances ne sont pas estimées à l'échelle de l'exemple mais à celle du *batch*.

Concrètement, lors de la phase d'apprentissage, la couche de normalisation conserve une estimation $\hat{\mu} \in \mathcal{I}_{1,1,D}$ de la moyenne de la distribution de chaque canal des données qui lui sont présentées. Il en va de même pour la variance dont l'estimation est notée $\hat{\sigma} \in \mathcal{I}_{1,1,D}$. Chaque nouveau *batch* arrivant entraine un ajustement de ces deux paramètres. En notant $C_1, \ldots, C_B \in \mathcal{I}_{W,H,D}$ les cartes de caractéristiques en entrée de la couche associée aux membres du *batch* considéré, les règles de mises à jour des estimations sont les suivantes :

$$\hat{\mu} \leftarrow (1-M)\hat{\mu} + M\bar{C} \tag{3.27}$$

$$\hat{\sigma} \leftarrow (1-M)\hat{\sigma} + M\mathrm{var}(C) \tag{3.28}$$

où, pour tout $z \in [\![1; D]\!]$,

$$\bar{C}_z = \left(\frac{1}{BWH} \sum_{i=1}^{N} \sum_{j=1}^{W} \sum_{k=1}^{H} C_i(j,k,z) \right) \tag{3.29}$$

et

$$\mathrm{var}(C)_z = \left(\left(\frac{1}{BWH} \sum_{i=1}^{N} \sum_{j=1}^{W} \sum_{k=1}^{H} C_i(j,k,z)^2 \right) - \bar{C}_z^2 \right) \tag{3.30}$$

La variable M est un coefficient d'inertie et est fixée dans notre cas à 10^{-1}. Les auteurs de cette méthode [68] ont proposé que cette étape soit suivie d'une transformation affine via l'ajout de deux paramètres $\gamma \in \mathcal{I}_{1,1,D}$ et $\beta \in \mathcal{I}_{1,1,D}$, appris par descente de gradient stochastique. La couche de *batch normalization* calcule alors la fonction :

$$\begin{aligned} \mathcal{I}_{W,H,D} &\to \mathcal{I}_{W,H,D} \\ f &\mapsto \gamma \frac{f - \hat{\mu}}{\sqrt{\hat{\sigma} + \varepsilon}} + \beta \end{aligned} \tag{3.31}$$

où $\varepsilon = 10^{-5}$ garantit la stabilité numérique. Notons que les opérations s'appliquent à chacun des termes des vecteurs considérés. La raison motivant cette transformation affine réside dans sa capacité à compenser, si besoin, l'effet de la normalisation. En effet, il suffit que $\gamma = \sqrt{\hat{\sigma} + \varepsilon}$ et $\beta = \hat{\mu}$ pour que BN devienne la fonction identité. Ainsi, l'ajout d'une couche de *batch normalization* ne réduit pas l'expressivité du réseau, dans le sens où toute application

représentable par un réseau de neurones dépourvu de couche de *batch normalization* l'est également si l'on adjoint à ce réseau une ou plusieurs couches de cette nature.

Dropout Pour prévenir le sur-apprentissage, c'est-à-dire l'obtention d'un modèle trop spécifique aux données d'apprentissage, la technique de ***dropout*** [69] est parfois mise en œuvre. Il s'agit d'une méthode que l'on applique sur une couche complètement connectée ou convolutive et qui va désactiver certains des éléments la composant en mettant leur valeur de sortie à 0. Pour chaque élément, la neutralisation de ce dernier est déterminée par la réalisation d'une loi de Bernoulli dont le paramètre est fixé en amont. Un élément neutralisé a, ainsi, un gradient nul et sa valeur demeure alors inchangée lors de la mise à jour des poids pour le *batch* en cours.

De cette façon, chaque *batch* est appris sur un « sous-réseau » du modèle, correspondant à l'ensemble des neurones n'ayant pas été désactivés. Une fois la phase d'apprentissage terminée, le poids de chaque élément concerné par le *dropout* est alors multiplié par le paramètre de la loi de Bernoulli employée. Le *dropout* peut alors être vu comme une manière détournée de constituer un comité de réseaux de neurones tout en ne conservant qu'un seul réseau. Ainsi, cette méthode a un effet de régularisation sur l'apprentissage et limite les risques de sur-apprentissage, le sous-réseau appris étant différent pour chaque *batch*.

3.1.2 Architectures de classification

Un **classifieur** est un modèle statistique qui cherche à prédire la **classe**, c'est-à-dire la nature, de la scène représentée au sein d'une image à partir de cette dernière. Dans le cadre de la classification, seul un nombre fini de classes est considéré. Pour qu'un réseau convolutif puisse être utilisé en tant que classifieur, une façon de procéder est de placer autant de neurones dans la dernière couche complètement connectée qu'il y a de classes prises en compte et d'employer la fonction *softmax* comme fonction d'activation. Si cette dernière couche compte K neurones, on a

$$\mathrm{softmax}_K(z_1, \dots z_K) = \left(\frac{e^{z_1}}{\sum_{i=1}^{K} e^{z_i}}, \dots, \frac{e^{z_K}}{\sum_{i=1}^{K} e^{z_i}} \right) \tag{3.32}$$

La classe prédite par le modèle est alors $i \in [\![1; K]\!]$ tel que

$$z_i = \mathrm{argmax}\, \mathrm{softmax}_K(z_1, \dots z_K) \tag{3.33}$$

Pour le cas particulier où $K = 2$, il est possible de ne conserver qu'un seul neurone au sein de la dernière couche et de considérer la fonction sigmoïde comme fonction d'activation de ce dernier, cette fonction étant définie par

$$\begin{aligned} \mathrm{sigm} \colon \mathbb{R} &\to [0; 1] \\ x &\mapsto \frac{1}{1 + \exp(-x)} \end{aligned} \tag{3.34}$$

Si, à l'issue de cette fonction d'activation, le résultat obtenu est inférieur à $\frac{1}{2}$, la classe 1 est prédite. Dans le cas contraire, le résultat du classifieur est la classe 2.

Il existe de nombreuses d'architectures de classification. Nous ne présentons ici que celles qui ont été utilisées dans cette thèse (*cf.* figure 3.5).

3.1.2.1 LeNet

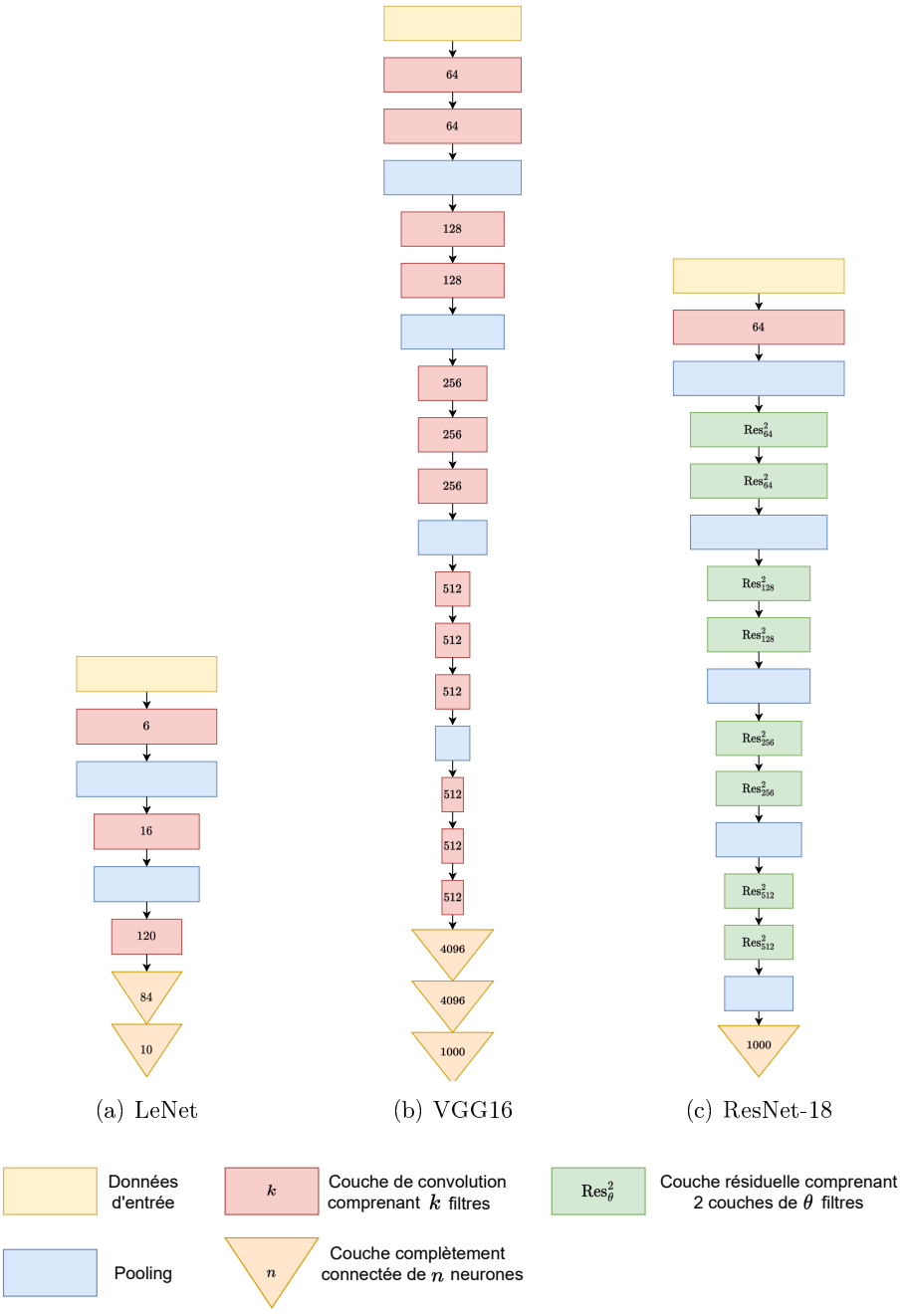
LeNet [70] est l'archétype du réseau convolutif présenté en figure 3.4. Il est composé de trois couches de convolution entrecoupées de deux couches d' *average pooling*. Un Perceptron multicouches est ajouté en sortie de cette dernière couche de convolution pour compléter le modèle. C'est la fonction tangente hyperbolique qui est utilisée comme fonction d'activation. La sous-figure 3.5a donne le détail de cette architecture.

3.1.2.2 VGG

Les architectures VGG [18] ont pour double particularité de n'utiliser que des filtres de convolution dont le noyau est de taille 3×3 et d'intercaler plusieurs couches de convolution entre des couches de *max pooling*. La fonction d'activation utilisée est alors la fonction ReLU. En faisant abstraction des biais et des fonctions d'activations, chaque groupement de k couches de convolution de noyau 3×3 a un champ réceptif (*i.e.* la taille de voisinage autour d'un point ayant une influence sur le résultat obtenu en ce dernier par application de la couche de convolution considérée) équivalent à celui d'une couche de convolution de noyau $(2k+1)\times(2k+1)$, tout en comprenant moins de paramètres. Or, un champ réceptif plus large permet de mieux prendre en compte les dépendances spatiales au sein des images d'entrée ou des cartes de caractéristiques. De cette façon, il est possible de réduire l'empreinte mémoire du réseau et ainsi de construire des architectures plus profondes. Le modèle que nous utilisons dans la thèse, VGG16, est détaillé dans la sous-figure 3.5b. Notons qu'il existe également une architecture VGG plus profonde, VGG19. Cette dernière demeure toutefois moins populaire que sa variante à 16 couches en raison d'une empreinte mémoire parfois jugée trop élevée au regard du gain de performances permis par les couches de convolution additionnelles de VGG19.

3.1.2.3 ResNet

L'architecture ResNet [4] vise à répondre à la problématique du dimensionnement des réseaux de neurones. Peu profonds, ces derniers ont des performances réduites. À l'inverse, s'ils comportent un grand nombre de couches de convolution, ils deviennent difficiles à apprendre. Pour obtenir des réseaux à la fois très profonds et ne posant pas de difficulté d'apprentissage, les auteurs de ResNet partent de l'observation suivante : si l'on dispose d'un réseau appris, il est possible de lui ajouter une couche de convolution sans pour autant modifier son comportement d'origine. En effet, il suffit que cette couche modélise la fonction identité. Dès lors, s'il existe un réseau d'une certaine profondeur



(a) LeNet (b) VGG16 (c) ResNet-18

FIGURE 3.5 – Classifieurs neuronaux. Pour LeNet, les couches de convolution ont un noyau de taille 5×5. Pour toutes les autres couches de convolution, le noyau est de taille 3×3 à l'exception de la première couche de convolution de ResNet, qui a un noyau 7×7.

résolvant la tâche considérée, tout réseau de profondeur supérieure peut théoriquement présenter les mêmes performances. Ainsi, une solution au problème de dimensionnement des réseaux de neurones serait de considérer des réseaux arbitrairement profonds.

Néanmoins, cette mécanique n'est pas évidente à réaliser puisqu'il n'existe qu'un seul jeu de paramètres permettant à une couche de convolution de représenter la fonction identité.

Par ailleurs, il est comparativement plus simple d'obtenir la fonction identiquement nulle par ajout d'une couche de convolution : il suffit de choisir ReLU comme fonction d'activation et choisir des paramètres de sorte que la sortie du neurone (ou de la couche de convolution) soit négative. En supposant que la carte de caractéristiques (ou l'image) en entrée, notée I, prenne toutes ses valeurs dans un intervalle $[m; M]$, et en notant C l'application d'une couche de convolution de noyau N et de biais B sur l'image d'entrée, on a, pour tout $(x, y, z) \in \mathbb{Z}$,

$$\begin{aligned} & \mathrm{ReLU}(C(x,y,z)) = 0 && (3.35)\\ \iff & C(x,y,z) \leq 0 && (3.36)\\ \iff & \sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{k=1}^{D} (N(i,j,k)I(x-i,y-j,z-k) + B(1,1,k)) \leq 0 && (3.37) \end{aligned}$$

Quelles que soient les valeurs prises par N, il suffit de poser, pour tout $k \in [\![1; D]\!]$,

$$B(1,1,k) \leq -M \left(\sum_{i=1}^{W}\sum_{j=1}^{H} |N(i,j,k)| \right) \tag{3.38}$$

pour que l'inégalité soit vérifiée. L'ensemble des paramètres permettant la modélisation de la fonction identiquement nulle contient donc un convexe non vide lorsque ReLU est prise pour fonction d'activation.

Forts de ces deux constats, les auteurs introduisent la notion de connexion résiduelle (voir figure 3.6). Ce composant, noté Res_θ^k par la suite, est constitué de k couches de convolution comportant chacune θ filtres de noyau 3×3 ainsi que d'un « court-circuit » représentant le résidu. Si on note $F(x)$ le résultat de l'application de ces k couches de convolution sur une entrée x, le résultat sur cette même entrée par l'ensemble de la connexion résiduelle vaut $F(x) + x$. Ainsi, pour que cet ensemble puisse modéliser la fonction identité, il faut et il suffit que $F(x)$ encode la fonction identiquement nulle. Bien que les propriétés énoncées précédemment ne soient valables que pour $k = 1$, les auteurs ont empiriquement constaté de meilleurs résultats avec $k = 2$.

Des travaux postérieurs [71] ont, en outre, montré que l'utilisation de connexions résiduelles au sein d'un architecture neuronale permettait d'obtenir une fonction de coût plus lisse que celles des architectures qui en sont dépourvues, telle que la famille des modèles VGG. Le processus d'optimisation des paramètres s'en voit alors facilité.

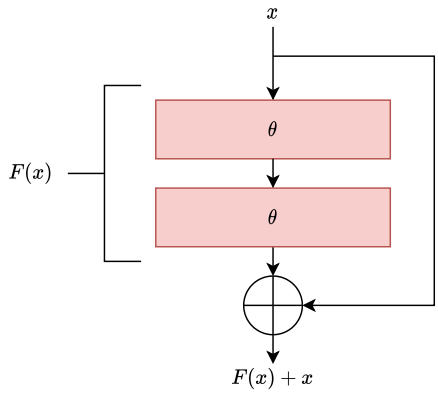


FIGURE 3.6 – Connexion résiduelle Res_θ^2 utilisée dans [4]

La sous-figure 3.5c présente l'architecture ResNet-18, que nous employons dans la thèse.

3.1.2.4 Interprétation des prédictions de classifieurs

Il n'est pas toujours évident de comprendre la logique interne des réseaux de neurones. C'est, tout particulièrement, le cas pour les classifieurs puisqu'une seule étiquette est prédite pour une image entière. Dès lors, on ne sait pas quel(s) élément(s) de l'image a (ont) conduit le classifieur à ce résultat.

Durant la dernière décennie, plusieurs méthodes ont été proposées pour tenter de répondre à cette question. Une des plus utilisées est Grad-CAM [19]. Cette approche consiste à établir, au niveau de la dernière carte de caractéristiques du classifieur (juste avant les couches complètement connectées), une carte d'activation en chaque point de ces cartes à l'aide des gradients qui y sont calculés et ce, pour chaque classe. On peut alors sur-échantillonner cette carte d'activation pour visualiser les éléments de l'image initiale qui contribuent à chacune des classes. En particulier, en considérant la carte d'activation pour la classe prédite on peut mettre au jour les éléments visuels ayant induit cette prédiction.

Les gradients sont utliisés pour mesurer, au niveau de la dernière couche de convolution, la contribution propre de chacun des filtres de convolution la composant. Formellement, en notant $CNN = P \circ F \circ E$ le réseau convolutif considéré (en retirant le softmax), E étant l'encodeur, F la fonction *flatten* et P le Perceptron multicouches qui lui est lié, l'importance du k-ème filtre de convolution pour la classe c et sur une image donnée I, noté α_k^c, correspond à la moyenne des gradients calculés pour chaque pixel de la carte de représentation formée par E par rapport à la classe c. Ainsi, on a, pour tout $k \in [\![1; D']\!]$,

$$\alpha_k^c = \frac{1}{W' \times H'} \left(\sum_{i=1}^{W'} \sum_{j=1}^{H'} \frac{\partial CNN_c}{\partial E(i,j,k)}(I) \right) \tag{3.39}$$

où W', H' sont les dimensions spatiales des cartes issues de l'encodeur $E(I)$, et D' leur nombre.

Pour obtenir la carte d'activation, il suffit alors de réaliser une moyenne pondérée sur la carte de représentation produite par E selon l'axe des canaux, en multipliant le k-ème canal par α_k^c. Une fonction ReLU est ensuite appliquée pour seuiller les activations. En notant GradCAM^c la carte d'activation de l'image I pour la classe c, il vient, pour tout $(i,j) \in \mathbb{Z}^2$,

$$\text{GradCAM}^c(i,j) = \text{ReLU} \left(\sum_{k=1}^{D'} \alpha_k^c E(I)(i,j,k) \right) \tag{3.40}$$

3.1.3 Modèles de cartographie

Il est possible de construire un modèle de cartographie par quadrillage régulier (tel que défini dans le chapitre introductif) à partir d'un classifieur. Pour ce faire, on découpe les images selon le quadrillage défini et on applique le classifieur sur chacune des cases ainsi formées. La figure 3.7 résume ce processus.

L'utilisation de classifieurs pour obtenir une cartographie par segmentation sémantique est possible mais présente plusieurs faiblesses. Parmi elles, un classifieur est un modèle prenant en entrée une image et qui produit une unique prédiction pour résultat. Pour obtenir une information de localisation associée à cette prédiction, nous pouvons avoir recours à un algorithme de fenêtre glissante. Néanmoins, cet algorithme est particulièrement lent : pour réaliser une cartographie définie au pixel près, il faut appliquer le modèle en tout point de l'image considérée. Il en résulte des coûts de calculs souvent prohibitifs.

Une des approches proposées a alors été de construire un modèle de segmentation sémantique intégralement neuronal. La première architecture réalisant une segmentation sémantique de façon complètement neuronale est FCN [72]

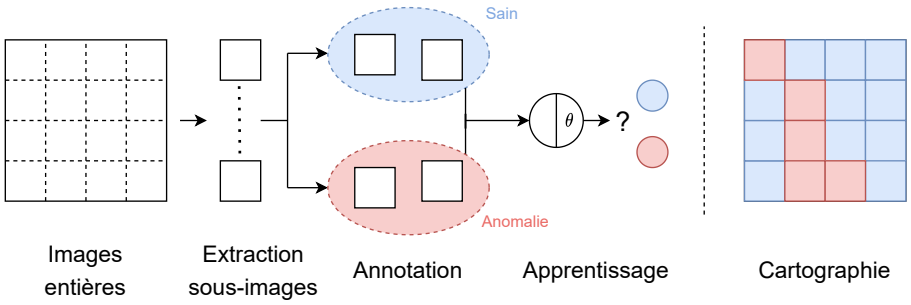


FIGURE 3.7 – Cartographie par quadrillage régulier à l'aide d'un classifieur

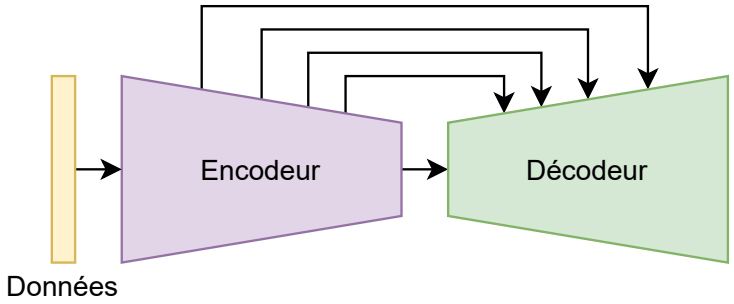
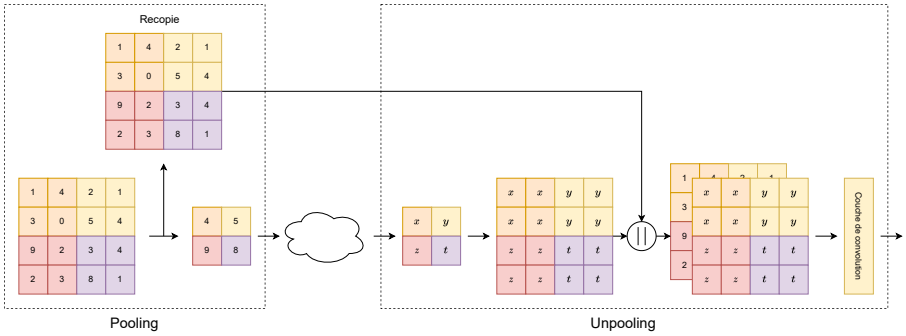


FIGURE 3.8 – Architecture de type encodeur/décodeur introduite par U-Net [5]

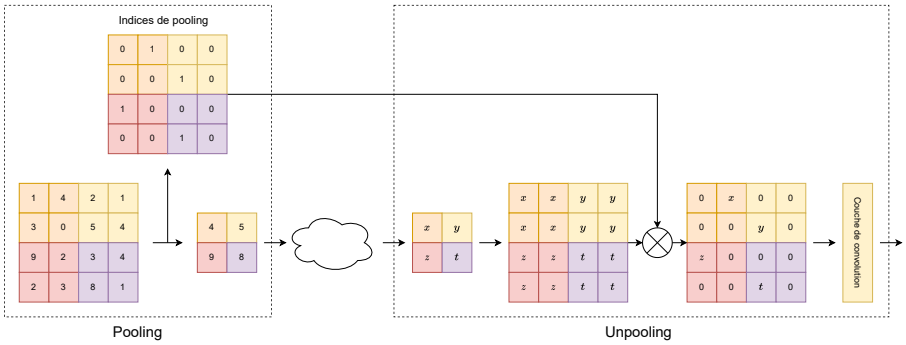
(pour *Fully Convolutional Network*), qui sur-échantillonne puis somme les résultats des trois dernières couches de convolution d'un même encodeur.

Un premier incrément par rapport à l'architecture FCN a été apporté par Ronneberger *et al.* [5] qui ont proposé U-Net. Dans cette architecture, l'encodeur est complété par un **décodeur**. Un décodeur est un composant neuronal visant à reconstruire une prédiction aux dimensions des données d'entrée. Il est de même composition qu'un encodeur mais les couches de *pooling* sont remplacées par des couches augmentant la dimension spatiale des cartes de représentation. Le décodeur de U-Net est « symétrique » par rapport à l'encodeur, c'est-à-dire qu'il comporte autant de couches de convolution que l'encodeur, ceci afin de permettre une meilleure prise en compte du contexte spatial dans l'apprentissage. La figure 3.8 illustre cette architecture. Notons la présence de courts-circuits qui favorisent, au sein du décodeur, la saillance des résultats obtenus par les couches de convolution au fur et à mesure que progresse leur résolution spatiale. Ces courts-circuits consistent, dans l'encodeur, à conserver une part de l'information perdue à chaque couche de *pooling* et ce, afin de la réinjecter dans le décodeur au moment de la reconstruction des résultats intermédiaires.

Dans [6], Badrinarayanan *et al.* proposent une modification de la nature des courts-circuits à travers le modèle SegNet. Là où U-Net recopie les résultats de filtres de l'encodeur pour les concaténer avec les cartes de caractéristiques sur-échantillonnées qui les précèdent, SegNet n'en reprend que la forme à travers des couches de *max unpooling* (la figure 3.9 montre les différences entre les deux approches). L'intérêt est double : outre une réduction de l'empreinte mémoire de l'architecture, ne reproduire que la forme des résultats des filtres lors de la reconstruction permet également d'isoler les caractéristiques de bas niveau de celles portant davantage de sémantique. On peut, dès lors, s'attendre à une meilleure généralisation puisque le décodeur est moins spécifique aux caractéristiques de l'encodeur et, en conséquence, aux données en entrée. La contrepartie de cette forme de courts-circuits est qu'elle produit, à sa sortie, des cartes de caractéristiques creuses, c'est-à-dire constituées majoritairement de 0.



(a) Court-circuit de U-Net, basé sur la recopie et la concaténation



(b) Court-circuit de SegNet, basé sur les indices de *pooling*

FIGURE 3.9 – Comparaison des mécanismes de court-circuit entre U-Net [5] et SegNet [6] sur un même exemple. Le symbole « || » représente la concaténation selon l'axe des canaux et $\otimes$ la multiplication terme à terme.

Les couches de convolutions les recevant en entrée doivent alors les « densifier », ce qui peut rallonger le temps d'apprentissage.

Bien d'autres modèles ont été développés depuis [73, 74, 75]. Cependant, malgré la profusion d'architectures de segmentation sémantique, les modèles U-Net et SegNet demeurent populaires dans les applications pratiques du fait de leur facilité de prise en main et de leurs bonnes performances. En effet, ces deux architectures sont simples à comprendre et donc facile à adapter. Il est ainsi possible de modifier le nombre de couches de convolution pour réduire le nombre de paramètres et le coût mémoire du modèle ou, à l'inverse, d'augmenter sa capacité de représentation.

3.1.4 Réseaux antagonistes génératifs

Les réseaux antagonistes génératifs [76], aussi appelés ***GAN*** (pour *Generative Adversarial Networks*), constituent une méthode pour la génération de

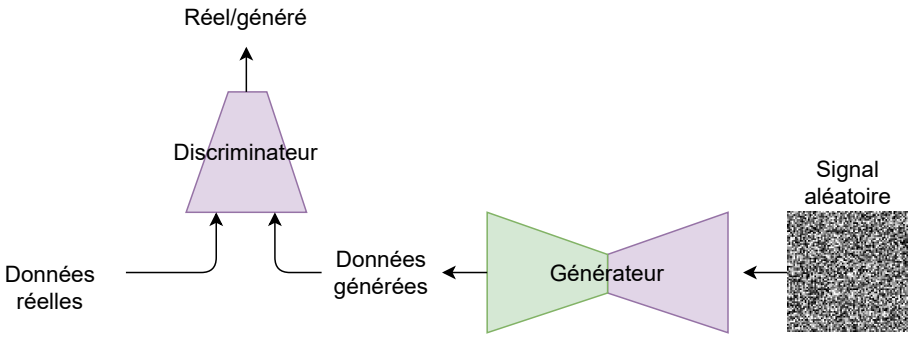


FIGURE 3.10 – Paire de réseaux antagonistes génératifs

données synthétiques. Si leur vocation première ne porte pas sur les méthodes de cartographie, ils peuvent cependant y contribuer de manière plus ou moins directe (des exemples d'application de réseaux antagonistes génératifs pour notre problématique applicative sont donnés en chapitre 4). Un *GAN* est composé d'une paire de réseaux de neurones poursuivant des objectifs antagonistes (voir figure 3.10).

Le premier réseau, appelé **générateur**, est entraîné à générer des exemples semblables à un jeu de données à partir de signaux aléatoires. Le réalisme des exemples générés est guidé par un second réseau, appelé **discriminateur**, qui est entraîné à faire la distinction entre les exemples produits par le générateur et ceux utilisés pour l'apprentissage de ce dernier.

L'apprentissage de ces deux réseaux doit alors être effectué conjointement. Il en résulte une certaine instabilité, puisque le « critère » d'apprentissage du générateur est lui-même modifié pendant l'apprentissage. Ainsi, si le discriminateur est trop performant, le générateur aura du mal à s'améliorer lors des époques suivantes. À l'inverse, si le discriminateur est peu efficace, le générateur n'est pas incité à produire des images réalistes. Réussir à faire converger une paire de réseaux antagonistes génératifs vers une configuration dans laquelle le générateur produit des exemples réalistes et inédits représente donc une difficulté.

3.1.5 Apprentissage multi-modal

L'utilisation de données multi-modales est une problématique importante en apprentissage profond. En effet, contrairement aux images photographiques (représentées en RGB), qui sont composées de trois canaux de sémantique et de dynamiques proches, les images multi-modales peuvent être constituées de canaux représentant des grandeurs physiques diverses (infra-rouges, distance au capteur, lumière polarisée, etc.), prenant ainsi leurs valeurs dans des intervalles différents, et n'ayant possiblement pas tous la même résolution. La question est alors de savoir comment fusionner ces canaux aux caractéristiques hétérogènes

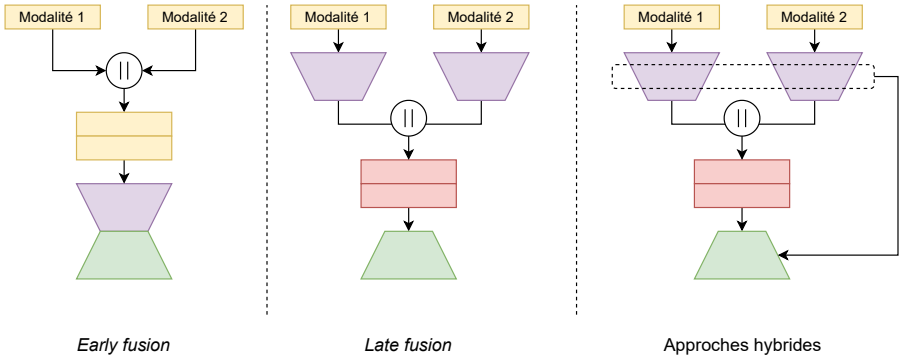


FIGURE 3.11 – Illustration des trois stratégies de fusion de données avec deux modalités (d'après [7]). Le symbole « || » représente la concaténation selon l'axe des canaux.

au sein d'une architecture neuronale pour exploiter au mieux l'information disponible.

Dans une revue de littérature, Zhang *et al.* [7] recensent trois stratégies possibles pour répondre à cette question : *early fusion*, *late fusion* et les approches hybrides (*cf.* figure 3.11). Cet article s'intéresse plus spécifiquement aux méthodes de segmentation sémantique mais les approches proposées sont directement transposables à la classification.

Early fusion La stratégie d'*early fusion* peut être vue comme l'approche naïve de l'apprentissage multi-modal. Elle consiste, en effet, à concaténer les différentes modalités en amont du réseau de neurones. Hormis la profondeur des filtres de la première couche, l'architecture de ce dernier n'a alors pas besoin d'être adaptée.

Cette stratégie peut s'avérer payante en termes de résultats mais elle demande d'apporter un soin particulier à la normalisation et au pré-traitement des différentes modalités. Il faut, à ce titre, veiller à ce que les valeurs de chacune d'elles soient du même ordre de grandeur tout en s'assurant pour que les éléments d'intérêt demeurent saillants. De plus, il est nécessaire que toutes les modalités aient la même résolution au moment de les présenter au réseau de neurones, condition qui n'est pas remplie pour tous les capteurs.

Late fusion Dans la stratégie de *late fusion*, chaque modalité dispose de son propre encodeur. Les représentations qui en sont issues sont ensuite fusionnées, le plus souvent par concaténation. Un décodeur est ensuite positionné après cette étape de fusion pour générer la prédiction. Comme les modalités ne sont entremêlées qu'à travers leurs représentations, il n'est plus nécessaire de trouver une normalisation commune aux différentes modalités. Il devient, par ailleurs, possible de travailler avec des modalités de résolutions distinctes.

En contrepartie de la souplesse qu'elle permet, cette approche induit un coût mémoire plus important. En effet, les architectures qui en relèvent présentent autant d'encodeurs que de modalités considérées. Or, chacun de ces encodeurs doit être chargé en mémoire, préférentiellement sur la mémoire d'une carte de calcul. Ainsi, les architectures adoptant cette stratégie de *late fusion* ont tendance à avoir une empreinte mémoire importante.

Stratégies hybrides Certains travaux ont étendu l'idée de la *late fusion* en intégrant, dans la phase de décodage, des courts-circuits en provenance d'un ou plusieurs encodeurs. L'objectif de ces approches est de parvenir à tirer le meilleur des deux mondes : la souplesse de la *late fusion* sur les données d'entrée tout en conservant la possibilité d'extraire des caractéristiques de bas niveau communes aux deux modalités propres à la *early fusion*.

Approche retenue Nous avons opté pour la stratégie d'*early fusion* pour la combinaison des différentes modalités des données LCMS, les images d'intensité et de profondeur étant de même résolution et, de surcroît, parfaitement recalées. Outre le fait qu'elle constitue l'approche la plus rapide à mettre en œuvre, elle présente aussi l'avantage d'être économe en mémoire. Les autres approches, si elles peuvent témoigner de bons résultats, demandent un temps de mise au point plus important. Néanmoins, comme mentionné auparavant, ce choix nous contraint à faire preuve de vigilance quant à la normalisation des données.

3.2 Méthodologie d'évaluation

L'évaluation consiste à mesurer l'adéquation entre les prédictions d'un modèle sur des données et les vérités terrain associées à ces dernières. Nous avons mis en œuvre deux types d'évaluation :

— Une évaluation quantitative, tout d'abord, pour laquelle on jauge les performances des modèles à travers diverses métriques couramment utilisées dans la littérature. Ces métriques nous renseignent ainsi sur le comportement du modèle évalué (sur-détection, sous-détection, etc.). Elles permettent également de comparer deux approches lorsque ces dernières sont évaluées sur la même base d'exemples.

— Une évaluation qualitative, ensuite, dans laquelle des exemples de prédictions sont commentés. On cherche ainsi à identifier des situations types dans lesquelles il présente de bons résultats et, à l'inverse, celles pour lesquelles les performances sont plus faibles. L'évaluation qualitative peut représenter une première étape dans l'évaluation opérationnelle des modèles. En particulier, on peut estimer, sur les exemples de prédictions, l'ampleur du travail manuel qu'il faudrait réaliser pour corriger la prédiction.

3.2.1 Métriques

Pour évaluer quantitativement un modèle de cartographie sur un jeu de données, on utilise des métriques. Les cartographies que nous considérons sont binaires, c'est-à-dire que les vérités terrain, et donc les prédictions, ne peuvent admettre que deux valeurs : 0 pour la classe négative (revêtement sain) et 1 pour la classe positive (revêtement présentant une anomalie). Un grand nombre de ces indicateurs repose sur une matrice appelée **matrice de confusion**. Cette matrice synthétise la performance du modèle en comptabilisant, pour chaque classe, la quantité d'exemples issus de cette classe qui sont correctement prédits et ceux pour lesquels le modèle a fourni une réponse erronée. Pour la cartographie par quadrillage régulier, un exemple correspond à un maillage de ce quadrillage. Quant à la cartographie au pixel près, chaque exemple équivaut à un pixel. Dans le cas binaire, la matrice de confusion est une matrice 2×2, comprenant alors quatre coefficients :

les vrais négatifs (TN) : Nombre de revêtements sains classés comme revêtements sains

les faux positifs (FP) : Nombre de revêtements sains classés comme anomalies

les faux négatifs (FN) : Nombre d'anomalies classées comme revêtements sains

les vrais positifs (TP) : Nombre d'anomalies classées comme anomalies

Comme il n'y a pas de convention universelle quant à l'ordonnancement de ces coefficients au sein de la matrice, nous les arrangeons de la façon suivante

$$\begin{pmatrix} \mathrm{TN} & \mathrm{FP} \\ \mathrm{FN} & \mathrm{TP} \end{pmatrix} \tag{3.41}$$

La première ligne correspond donc aux exemples dont la vérité terrain est 0 (classe saine) tandis que la seconde est associée aux exemples comprenant des anomalies.

La performance du modèle est d'autant meilleure que la matrice est proche d'une matrice diagonale. Une des informations que l'on peut extraire de cette matrice est l'**exactitude**, notée E, qui correspond à la proportion d'éléments sur cette diagonale. On a

$$E = \frac{TN + TP}{TN + FP + FN + TP} \tag{3.42}$$

Une des limites de cet indicateur est qu'il ne tient pas compte de la fréquence d'apparition des différentes classes. Ainsi, si une classe est largement majoritaire par rapport à une autre, ce qui est le cas dans notre domaine applicatif, les anomalies étant bien plus rares que les revêtements sains, la performance réalisée par la classe sur-représentée devient prépondérante dans l'indicateur.

C'est pourquoi une autre métrique, l'**exactitude pondérée**, notée EP, lui est généralement préférée :

$$EP = \left(\underbrace{\frac{TN}{TN+FP}}_{S} + \underbrace{\frac{TP}{FN+TP}}_{R} \right) \Big/ 2 \tag{3.43}$$

où S est appelé **spécificité** et correspond à la proportion d'éléments sains qui sont effectivement classés comme sains tandis que R est le **rappel** et mesure la même grandeur, mais pour les anomalies. Notons que, lorsque qu'il y a autant d'éléments dans chaque classe, on a $TN + FP = FN + TP$, et donc $EP = E$.

Une autre métrique fréquemment calculée est la précision P, qui correspond à la proportion d'éléments d'intérêt parmi les détections d'anomalies :

$$P = \frac{TP}{FP+TP} \tag{3.44}$$

Le F_1-score est régulièrement utilisé comme synthèse de la précision et du rappel et consiste en leur moyenne harmonique :

$$F_1 = \frac{2}{\frac{1}{R} + \frac{1}{P}} \tag{3.45}$$

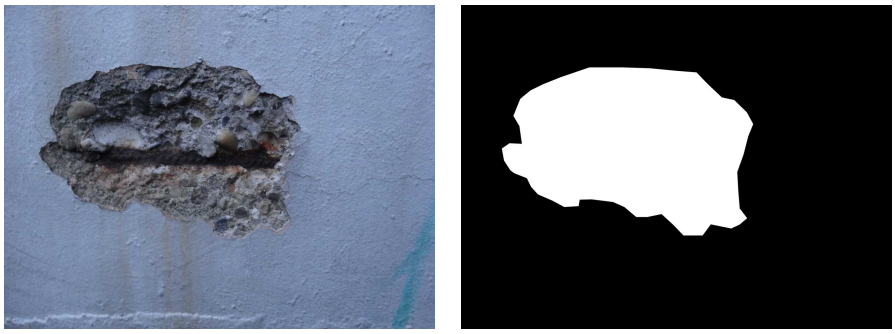
Une autre métrique parfois employée est l'**IoU** (pour *Intersection over Union*), également appelée **indice de Jaccard**. Elle est définie par

$$IoU = \frac{\#\left(\{\text{Vérité terrain « anomalie »}\} \cap \{\text{Prédiction « anomalie »}\}\right)}{\#\left(\{\text{Vérité terrain « anomalie »}\} \cup \{\text{Prédiction « anomalie »}\}\right)} \tag{3.46}$$

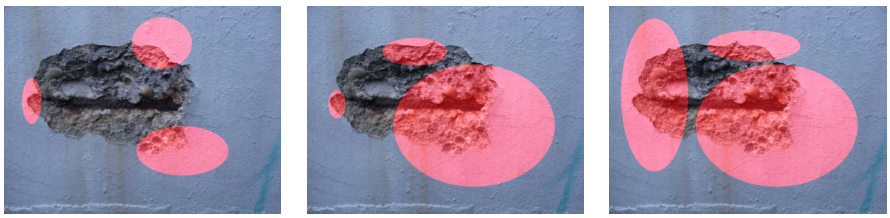
$$= \frac{TP}{FP+TP+FN} \tag{3.47}$$

Bien d'autres métriques peuvent être déduites de la matrice de confusion, cette liste ne recense que les plus utilisées dans la littérature.

Pour la segmentation sémantique, nous introduisons un dernier outil d'évaluation : le rappel par composantes connexes. Il permet de répondre à un écueil du rappel pour la segmentation sémantique. En effet, ce dernier est calculé à l'échelle du pixel. Or, pour certaines anomalies, on sait que la vérité terrain est constituée de larges composantes connexes, chacune représentant une anomalie distincte. Il est alors plus intéressant, d'un point de vue opérationnel, de compter le nombre d'anomalies effectivement détectées. Pour chaque composante connexe de la vérité terrain représentant une anomalie, on calcule la proportion des pixels y appartenant et prédits par le modèle comme anomalies. Si cette proportion est supérieure à un certain seuil, que nous appellerons **seuil d'admission** par la suite, la composante connexe est considérée comme détectée. Le rappel par composantes connexes est alors défini comme la fonction qui associe, à chaque seuil d'admission, la proportion de composantes connexes



(a) Exemple (b) Vérité terrain



(c) $T = 12.35\%$ (d) $T = 60.55\%$ (e) $T = 80.73\%$

FIGURE 3.12 – Illustration du rappel par composantes connexes sur un exemple présenté dans la rangée du haut (image à gauche, masque de vérité terrain associé à droite). Dans la rangée du bas, trois exemples de prédictions sont présentés (les pixels prédits comme fers apparents apparaissent en rouge). Le seuil d'admission T en dessous duquel la composante connexe est considérée comme détectée est renseigné en légende de chacun de ces exemples.

détectées selon ce seuil. La figure 3.12 illustre cette notion de rappel par composantes connexes à travers un exemple.

On pourrait également calculer la précision par composantes connexes mais la sémantique d'une telle métrique est moins évidente. En effet, pour le rappel, nous comptons les composantes connexes de la vérité terrain. Pour la précision, il faudrait, cette fois-ci, dénombrer les composantes connexes de la prédiction. Or, le réseau de neurones peut générer un grand nombre de petites composantes connexes éparses, sans que la taille de ces composantes ne renseigne sur leur nature (vrai ou faux positif). On ne retrouve donc pas la même richesse sémantique dans la précision par composantes connexes. Cette métrique n'a donc pas été retenue.

3.2.2 Influence de la composition du jeu de données sur les métriques

Dans cette section, nous démontrons que le rappel, la sensibilité et la précision ne dépendent pas uniquement des performances du modèle mais varient

aussi selon la composition (proportion d'exemples de chaque classe) de ce jeu de données. On note $\mathcal{N} = TN + FP$ le nombre total d'éléments de la classe négative (*i.e.* revêtements sains), $\mathcal{P} = TP + FN$ le nombre total d'éléments de la classe positive (*i.e.* revêtements présentant une anomalie) et $n = \mathcal{N} + \mathcal{P}$ le nombre total d'éléments, toutes classes confondues. On définit, de plus, α comme la proportion d'exemples appartenant à la classe négative. Ainsi, on a $\alpha = \frac{\mathcal{N}}{n}$. De même, $1 - \alpha = \frac{\mathcal{P}}{n}$.
Il en ressort que :

$$\frac{\alpha}{1-\alpha} = \frac{\mathcal{N}}{\mathcal{P}}. \tag{3.48}$$

Par définition, $R = \frac{TP}{\mathcal{P}}$ et $S = \frac{TN}{\mathcal{N}}$, i.e. $1 - S = \frac{FP}{\mathcal{N}}$, puisque $\mathcal{N} = TN + FP$. On en déduit que :

$$\frac{1-S}{R} = \frac{\mathcal{P}}{\mathcal{N}}\frac{FP}{TP}. \tag{3.49}$$

Reprenons maintenant la définition générale de la précision :

$$P = \frac{TP}{TP + FP}, \tag{3.50}$$

qui s'écrit aussi :

$$P = \frac{1}{1 + \frac{FP}{TP}} = \frac{1}{1 + \frac{\mathcal{N}}{\mathcal{P}}\frac{\mathcal{P}}{\mathcal{N}}\frac{FP}{TP}}. \tag{3.51}$$

En utilisant les expressions (3.48) et (3.49), on obtient :

$$P = \frac{1}{1 + \frac{\alpha}{1-\alpha}\frac{1-S}{R}} \tag{3.52}$$

Il existe donc une relation fonctionnelle entre les indicateurs suivants :

$$P, S, R, \alpha \tag{3.53}$$

Il ressort de ce raisonnement que la connaissance de trois des quatre éléments cités dans 3.53 permet d'en inférer le quatrième. Or, puisque la fonction de coût que nous employons pour l'apprentissage de nos modèles (entropie croisée binaire et pondérée) pénalise uniformément les classes « saine » et « anomalie », les quantités S et R tendent à être élevées et à admettre des valeurs proches, induisant une valeur basse pour le terme $\frac{1-S}{R}$. En conséquence, lorsque α avoisine 0 (classe « anomalie » largement majoritaire), le terme $\frac{\alpha}{1-\alpha}$ tend vers 0. La précision s'en voit alors tirée vers le haut, et inversement lorsque α est proche de 1. Cette seconde situation étant celle que l'on rencontre dans la plupart de nos jeux de données, puisque les anomalies y sont minoritaires, les précisions obtenues par les modèles évalués sur ces derniers (et les F_1-scores associés) auront tendance à être bas.

Notons toutefois que cette conclusion n'est valable que lorsque la fonction de coût utilisée est l'entropie croisée binaire et pondérée et lorsque le biais de domaine entre les jeux d'apprentissage et de test est négligeable. Précisons

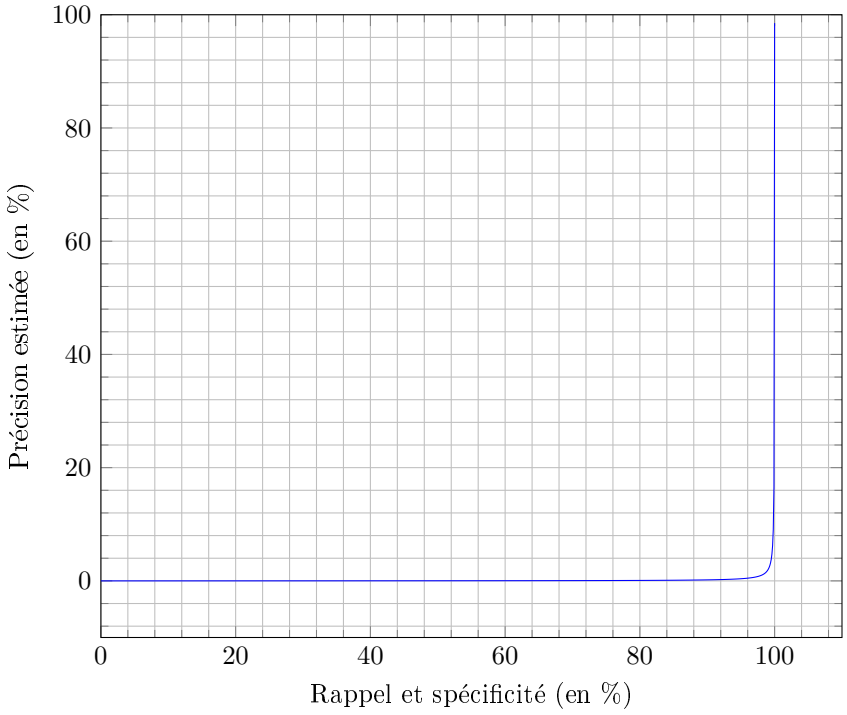


FIGURE 3.13 – Précision donnée par l'équation 3.52 en fonction du rappel et de la spécificité (supposés égaux pour les besoins de la visualisation) pour $\alpha = 0.9998$.

également que cette analyse n'interdit en rien l'obtention de modèles dont la précision est arbitrairement élevée. Elle indique seulement que de tels modèles doivent avoir un rappel et une spécificité remarquablement proches de 100%. La figure 3.13 présente la précision donnée par l'équation 3.52 en fonction du rappel et de la spécificité lorsque ces deux derniers indicateurs sont égaux et lorsque $\alpha = 0.9998$. Cette valeur de α correspond à la proportion de pixels représentant des fissures pour le jeu de test du tunnel de la Grand Mare (*cf.* chapitre 2). Il en ressort, par exemple, qu'un rappel et une spécificité s'établissant autour de 99% induirait une précision de l'ordre de 4%.

Conclusion du chapitre

Dans ce chapitre, nous avons fait un tour d'horizon des fondements de l'apprentissage profond ainsi que de la méthodologie d'évaluation employée plus largement dans le cadre de l'apprentissage automatique. Outre la présentation de ces différents aspects, nous avons motivé les choix réalisés pour la thèse. Il en ressort les éléments suivants :

- Les réseaux de neurones sont des modèles statistiques puissants, dans le sens où ils peuvent approcher avec une précision arbitraire toute fonction continue à valeur réelle et définie sur un compact de $\mathbb{R}^n$.
- Loin de constituer une chaîne d'opérations homogènes, ils sont composites, chaque composant répondant à une problématique définie.
- Les réseaux sont appris de façon itérative, en leur présentant, tour à tour, des exemples regroupés par *batch* et en comparant le résultat produit et celui qui était attendu.
- Ils peuvent être utilisés de plusieurs façons pour produire les cartographies d'anomalies. Nous en avons exploré deux : une première approche par classification et une seconde par segmentation sémantique.
- Plusieurs stratégies sont possibles pour gérer la multi-modalité des données : *early fusion*, *late fusion* et les stratégies hybrides. Dans cette thèse, nous n'explorerons que la première des trois.
- Il existe deux approches complémentaires pour évaluer les performances d'un réseau de neurones. Une approche qualitative, dans laquelle des exemples de prédictions sont analysés et une approche quantitative, pour laquelle on établit des métriques visant à mesurer l'écart entre la prédiction du modèle sur un exemple et la vérité terrain associée à cet exemple.
- Plus la classe d'intérêt est minoritaire, plus est difficile d'obtenir un modèle caractérisé simultanément par une précision haute et un rappel élevé.

Chapitre 4

État de l'Art : détection des anomalies au sein de structures de génie civil

Dans ce chapitre, nous mettons en avant les travaux les plus significatifs de la communauté scientifique qui visent à cartographier les anomalies au sein de structures de génie civil (non limitées aux ouvrages d'art). Travaillant à la fois sur des images photographiques et des relevés LCMS, nous considérons les contributions utilisant ces deux types de données, en étendant la catégorie des relevés LCMS aux données admettant une composante de profondeur, ci-après désignées par « données 3D ».

Les méthodes présentées sont regroupées en trois catégories : une première partie présente les approches sans apprentissage, une deuxième explore les stratégies avec apprentissage pour laquelle les auteurs font l'hypothèse que le biais de domaine entre les jeux d'apprentissage et de test est négligeable (nous parlons **d'apprentissage intra-domaine**), une troisième regroupe les approches avec apprentissage pour lesquels leurs auteurs mettent explicitement en œuvre une stratégie pour prendre en compte le biais de domaine, alors supposé significatif (nous parlons **d'apprentissage inter-domaines**).

Sommaire

4.1 Méthodes sans apprentissage (fissures)

Dans le cas des images photographiques, il existe une vaste littérature sur la segmentation sémantique des fissures ne reposant pas sur des mécanismes d'apprentissage mais sur des connaissances expertes. À notre connaissance, les fissures constituent la seule anomalie pour laquelle de telles méthodes ont été mises en œuvre. Ceci peut s'expliquer par l'importance que revêtent les fissures dans notre contexte opérationnel et par la (relative) simplicité géométrique qui les caractérise. Plus spécifiquement, l'hypothèse est faite que les fissures sont, visuellement parlant, constituées de pixels plus sombres que le reste de l'image et regroupées au sein de composantes connexes oblongues. Cette dernière propriété est parfois reformulée en disant que le relevé d'une fissure peut s'écrire comme la réunion de segments discrets, ayant chacun une épaisseur propre. Dans une revue de littérature datant de 2010 [77], Wang et Huang présentent quatre familles de méthodes proposées pour réaliser la segmentation de fissures au sein d'images en niveaux de gris. Nous les reprenons ci-après, en y adjoignant une autre famille de méthodes : les approches géométriques.

Approches par seuillage Les approches par seuillage [78, 79] opèrent en deux temps : une première passe consiste à nettoyer l'image (réduction du bruit, ajustement de la luminosité et du contraste, etc.) et une seconde réalise le relevé de la fissure dans l'image corrigée en seuillant l'image en fonction des niveaux gris de cette dernière.

Approches morphologiques Les approches morphologiques [8, 80] reposent sur les opérateurs de morphologie mathématique [81]. L'intérêt des opérateurs de morphologie mathématique réside dans leur capacité à « filtrer » facilement les différents éléments d'une image en faisant varier l'élément structurant employé. La figure 4.1 illustre cet aspect pour la détection des joints et des jonctions entre tuyaux. De plus, en travaillant avec des éléments structurants, ces approches permettent une meilleure gestion de la connexité du relevé de la fissure, là où les approches par seuillage traitent indépendamment chaque pixel.

Approches par percolation Les approches par percolation [82, 83] constituent des variantes de l'algorithme du germe [84] dans laquelle on fait itérativement croître une région au sein d'une image en faisant dépendre la vitesse de propagation de cette région de l'intensité des pixels traversés (la propagation est plus rapide pour les zones sombres de l'image). Si la circularité de la région ainsi construite est basse, cette dernière est considérée comme étant le relevé d'une fissure. Dans le cas contraire, cette région est considérée comme revêtement sain. Il suffit alors d'appliquer ce procédé en chaque point de l'image pour relever l'intégralité des fissures qui y figurent.

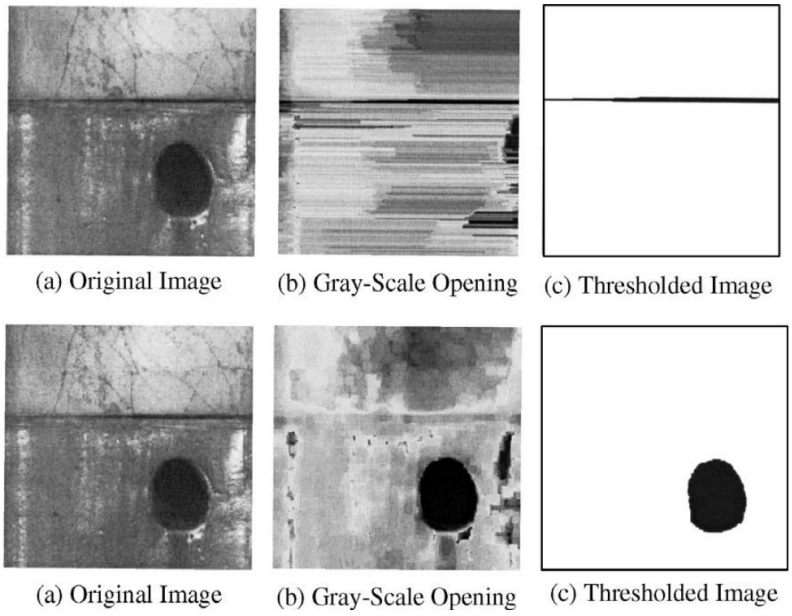


FIGURE 4.1 – Détection des joints et des jonctions entre tuyaux dans [8]. La première ligne montre le résultat d'une ouverture morphologique par un segment horizontal sur un exemple (détection du joint), la seconde montre le résultat d'une ouverture morphologique sur ce même exemple avec un élément structurant circulaire (détection d'une jonction entre tuyaux). L'image est acquise depuis l'intérieur d'un tuyau (source : [8]).

Approches semi-automatiques Afin d'améliorer la qualité des relevés des fissures, certaines approches proposent de guider ce processus par un opérateur chargé de cibler grossièrement les fissures. En plus de la méthode [50, 51] reposant sur le *fast marching* présentée dans le chapitre 2, plusieurs travaux se sont concentrés sur cet axe [85, 86]. Ces méthodes permettent d'avoir un relevé précis des fissures, mais demandent l'intervention d'un opérateur, ce qui n'est pas envisageable sur le plan opérationnel lorsque des ouvrages entiers seront à analyser. Elles peuvent cependant être mises à profit pour faciliter le travail d'annotation.

Approches géométriques Les approches géométriques consistent à détecter les points saillants des fissures dans un premier temps (extrémités, croisements entre deux fissures, etc.) avant de déterminer des chemins les reliant de sorte à reconstituer le relevé des fissures.

Dans [87, chapitre 2], une stratégie de détection des fissures par points d'intérêt est mise en œuvre. Deux types de points d'intérêt sont recherchés pour la segmentation des fissures : les points de type « micro-lignes », qui correspondent aux pixels susceptibles d'être traversés par une fissure, et les

points de type « coins », qui matérialisent leurs extrémités. Une fois ces points déterminés, l'orientation privilégiée autour des points micro-lignes est estimée avant qu'une reconstruction de la fissure ne soit réalisée par une variante de l'algorithme LSD (*Line Segment Detector*) [88].

La méthode MPS (*Minimal Path Search*) [89] repose également sur ce principe. Après une détection des points significativement sombres dans une image par un seuillage adaptatif, l'algorithme de Dijskstra est appliqué pour déterminer les chemins composés de pixels de faible intensité reliant ces différents points. Après suppression des chemins dont l'intensité moyenne est supérieure à un seuil, un post-traitement pour corriger la topologie (la méthode peut induire la présence de boucles, notamment) et l'épaisseur des relevés est employé. Il convient de souligner que l'algorithme de Dijkstra employé pour relever le tracé des fissures est structurellement proche de celui utilisant sur le *fast marching* que nous avons mis en œuvre pour obtenir les annotations de fissures.

Synthèse des approches sans apprentissage Les méthodes présentées dans cette section ont pour avantage de reposer sur des paramètres facilement interprétables. Il est donc relativement simple de prédire le comportement qu'auront les algorithmes lorsque ces paramètres sont modifiés.

Néanmoins, il est très difficile, si ce n'est impossible, de déterminer un bon jeu de paramètres permettant à ces différentes approches d'être robustes à la variabilité d'apparence des fissures et des revêtements sains environnants. L'utilisation de méthodes d'apprentissage pour la conception de modèles de détection apparaît alors comme nécessaire. Les approches sans apprentissage demeurent toutefois intéressantes pour la création de banques de données annotées qui pourront ensuite être employées au sein de méthodes d'apprentissage. En dehors de cette dernière application, nous n'avons pas mis en œuvre d'approches sans apprentissage dans nos travaux.

4.2 Apprentissage intra-domaine

Les approches avec apprentissage cherchent à répondre aux limites des méthodes précédemment citées : plutôt que de reposer sur des connaissances préalables ou une intervention humaine, un mécanisme d'apprentissage est mis en œuvre pour estimer les paramètres d'un modèle sur un ensemble de données, le plus souvent annotées. Dans cette section, nous considérons les approches pour lesquelles l'hypothèse est faite que le biais de domaine entre les jeux d'apprentissage et de test est négligeable.

4.2.1 Images photographiques

Utilisation de caractéristiques définies manuellement Certains modèles statistiques non neuronaux (forêts aléatoires [34], *Support Vector Machine* [90], etc.) n'opèrent pas directement sur les niveaux de gris des images mais les considèrent par le biais de caractéristiques qui en sont extraites (contours, en-

tropie, histogramme de gradient orienté [91], etc.). Le choix des caractéristiques à extraire met en jeu des connaissances expertes sur les problèmes traités.

Dans [92], Munawar *et al.* dressent une liste de contributions importantes ayant été réalisées cette dernière décennie dans le domaine de la reconnaissance de fissures (classification à l'échelle de l'image, relevé au pixel près et estimation de propriétés physiques sur la base de photographie). En particulier, une comparaison entre les modèles statistiques non neuronaux et les réseaux de neurones convolutifs est réalisée. Si l'ensemble de ces méthodes témoignent de bons résultats, les auteurs concluent que les méthodes par apprentissage profond sont les plus utilisées ces dernières années. Même si les auteurs n'avancent pas d'hypothèse quant à cet engouement, on peut raisonnablement supposer que les bonnes performances générales des réseaux de neurones, la vaste documentation dont ils jouissent ainsi que la mise à disposition de modèles pré-entraînés par la communauté scientifique aient contribué à ce succès.

Dans le cadre du projet européen ROBO-SPECT, Makantasis *et al.* [93] ont proposé une stratégie consistant à enrichir les images d'entrée par plusieurs cartes de caractéristiques représentant chacune une information différente (contours, fréquences, entropie, texture et histogramme de gradient orienté). Un réseau convolutif, dédié à la classification, est ensuite entraîné sur ces images. Si les résultats quantitatifs sont relativement élevés (88,6% d'exactitude et de F_1-score), on peut cependant noter qu'aucune étude différentielle (*ablation study*) n'est réalisée, si bien qu'on ne peut pas conclure sur un éventuel bénéfice de l'ajout manuel de caractéristiques aux données d'entrée. À notre connaissance, une telle technique n'a pas été reproduite par la suite dans notre champ applicatif.

Le recours aux caractéristiques définies manuellement pour la reconnaissance des anomalies semble se raréfier au sein de la communauté scientifique, au profit des méthodes d'apprentissage profond, généralement jugées plus performantes dans le cadre du traitement d'images.

Reconnaissance des fissures par apprentissage profond Un point saillant de la littérature sur la reconnaissance des fissures est que ces dernières, étant des objets géométriquement simples, sont parfois confondues par les modèles statistiques avec certaines parties saines mais semblables du revêtement. Si ce phénomène concerne toutes les méthodes par apprentissage, il est encore plus important pour les architectures convolutives qui, pour des raisons matérielles, imposent souvent de découper et/ou de redimensionner les images d'entrée, réduisant ainsi la quantité d'information utilisable par le modèle. Par exemple, sur les structures en maçonnerie, certains travaux [94, 95] ont constaté que les joints entre moellons sont parfois considérés, à tort, comme des fissures. Dans cette dernière contribution [95], Dais *et al.* ont en partie résolu le problème en proposant une technique permettant de réduire le nombre de ces fausses alarmes. Leur méthode vise à sélectionner, au sein des modèles générés durant l'apprentissage, celui maximisant le F_1-score sur le jeu de validation. Procéder de la sorte permet de sélectionner un modèle ayant un rappel et une

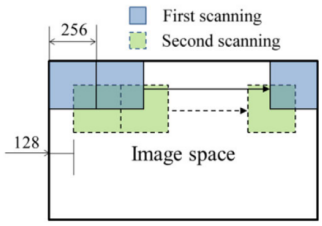


FIGURE 4.2 – Stratégie de détection (source : [9])

précision proches, ce qui représente souvent un bon compromis en cas de fort déséquilibre entre classes. De façon plus générale, Cha *et al.* [9] ont montré que le contexte entourant la fissure était capital pour la bonne détection de ces anomalies. En effet, si une aspérité est présente sur le bord d'une image, il n'est pas évident, en l'absence de contexte spatial, de savoir s'il s'agit d'une fissure ou d'un revêtement sain. La méthode proposée consiste alors à découper les images d'entrée selon deux quadrillages positionnés en décalage, de sorte qu'une fissure excentrée selon l'un des quadrillages ne le soit pas selon l'autre. La figure 4.2 présente les deux quadrillages utilisés.

Certains travaux [96, 25, 97, 98, 99, 100] évaluent l'intérêt d'architectures de segmentation sémantique conservant l'ensemble des représentations intermédiaires issues de l'encodeur pour les fusionner au moment du calcul de la prédiction. Se comparant systématiquement à des architectures de segmentation de type encodeur/décodeur, ces différentes études tendent à montrer la supériorité des approches par fusion hiérarchique. Celles-ci ont déjà été utilisée pour la reconnaissance de contours [101], ce qui peut expliquer les bonnes performances des modèles qui en découlent pour la reconnaissance des fissures. On peut cependant relever que toutes les fissures n'ont pas un contour marqué. De plus, peu d'éléments distracteurs, tels les joints, sont présents dans les données utilisées dans ces travaux. Bien que cette piste soit intéressante à explorer, il n'est pas garanti qu'elle améliore significativement les résultats pour notre domaine applicatif, dans lequel les représentations des parements de tunnels comprennent des joints, des passe-câbles et d'autres éléments visuels rectilignes ayant eux aussi une délimitation nette.

Des méthodes prenant en compte la topologie des relevés des fissures ou encore leur contexte spatial lors de l'apprentissage de réseaux de neurones ont été proposées dans la littérature. L'idée sous-jacente est que les fissures sont généralement assimilables à une forme de faible largeur, allongée et connexe. Entraînées à segmenter de fins objets, les architectures de segmentation risquent de prédire une succession de taches éparses et surdimensionnées au lieu des formes continues attendues. Li *et al.* [102] introduisent, dans la fonction de coût, un terme visant à pénaliser le défaut de connexité dans chaque 8-voisinage. La même idée a été proposée de façon indépendante par Mei *et al.* [103]. Les travaux de Pantoja-Rosero *et al.* [104] s'inscrivent dans cette même démarche et proposent, de plus, de convertir les vérités terrain en carte de distance ainsi que

d'ajouter un critère de connexité défini sur ces mêmes cartes de distance. Une autre approche, développée par Hu *et al.* [105], consiste à inciter le réseau à prédire des résultats ayant la même structure topologique que la vérité terrain. Ces méthodes présentent une plus-value importante pour les fissures complexes couramment rencontrées sur les revêtements de chaussée, tel le « faïençage » (*i.e.* réseau de fissures entrelacées et présentes en grand nombre dans une même zone). Néanmoins, les fissures présentes dans nos données admettent une topologie très simple, puisqu'elles sont généralement constituées d'une seule composante connexe. Les quelques tests que nous avons réalisés avec ces méthodes sur nos données n'ont pas permis de mettre en avant d'amélioration des résultats dans notre contexte.

Plus récemment, certains auteurs ont développé des méthodes prenant en compte le manque de représentativité de la classe « saine » au sein des jeux d'apprentissage. En effet, dans la plupart de ces jeux de données, l'accent est généralement mis sur la représentativité des fissures, ce qui peut dégrader les performances des modèles lorsque la classe « saine » du jeu à évaluer présente davantage de variabilité d'aspect que celle des jeux d'apprentissage. Pour ce faire, une première stratégie consiste à augmenter ces jeux d'apprentissages. Paleviius *et al.* [106] implémentent un tel procédé par le biais d'ajout d'ombres projetées sur des images de chaussées. Leur méthode consiste à supposer la chaussée planaire et à calculer, par une méthode de rendu (lancer de rayons), la surface projetée sur la chaussée par différents objets tridimensionnels (véhicules, constructions, végétations, etc.). Dans le cas des structures maçonnées, Loverdos *et al.* [107] cherchent à détecter explicitement les joints, qui sont des éléments saillants de ce type de construction et dont l'apparence peut visuellement les rapprocher des fissures. À cette fin, ils ont entraîné deux réseaux de neurones dédiés à la segmentation : un premier pour les fissures et un second pour les joints entre briques. Dans leur travail, ces deux modèles sont utilisés séparément mais l'apprentissage d'une architecture commune pour réaliser ces deux tâches est envisagé en perspective et pourrait permettre de réduire le nombre de fausses alarmes en incitant le modèle à avoir une meilleure compréhension des images présentant des revêtements maçonnés.

Reconnaissance des fers apparents par apprentissage profond Les travaux s'intéressant aux fers apparents sont plus rares que ceux visant à reconnaître les fissures. Cette sous-représentation de contributions peut s'expliquer par une sous-représentation de l'anomalie dans notre contexte opérationnel. En effet, les fers apparents ne concernent que les structure en béton armé quand les fissures peuvent affecter la plupart des matériaux de construction.

Santos *et al.* [108] ont mis en œuvre une méthode de cartographie par quadrillage régulier pour la détection des fers apparents. Si la méthode permet une annotation rapide des données, puisque réalisée à l'échelle de la sous-image, elle souffre cependant du même problème d'insuffisance du contexte spatial que celui évoqué lors de la présentation des travaux de Cha *et al.* [9], à savoir qu'un fer apparent qui n'est représenté que partiellement sur un des bords de la sous-

image risque d'être difficile à identifier, y compris par un expert du domaine.

On peut également citer la contribution de Savino *et al.* [109], qui proposent une approche par segmentation sémantique. Dans ses travaux, l'équipe du professeur Prendinger [110, 111, 112] développe une stratégie similaire. Notons toutefois que, dans leurs données, les annotations des fers apparents sont réparties en plusieurs classes, les armatures étant séparées des pertes de matière, là où Savino *et al.* [109] attribuent le même label à ces deux éléments. Ce haut degré de détail au sein des annotations est repris par Kim *et al.* [113] qui réalisent une segmentation d'instance des fers apparents, c'est-à-dire qu'en plus de déterminer la nature de chaque pixel de l'image, le réseau est également entraîné à discerner les instances d'objets de même classe apparaissant dans l'image.

Si la séparation, au niveau des annotations, des différentes parties des fers apparents (pertes de matière dans une classe et armatures métalliques dans un autre) peut permettre d'obtenir des cartographies plus détaillées, une telle opération demande, en contrepartie, un travail de labellisation supplémentaire. Compte tenu de l'importance des verrous qu'il reste à investiguer dans notre contexte applicatif avant de pouvoir mesurer le bénéfice de ces annotations enrichies, il ne nous a pas semblé judicieux d'opter pour cette approche. Ainsi, nous avons choisi de n'avoir qu'une seule classe pour représenter les fers apparents, à l'instar des travaux de Savino *et al.* [109].

Construction d'une cartographie globale de l'ouvrage Quelques approches cherchent, en plus de générer une cartographie locale des images, à agréger ces différentes cartes pour obtenir une représentation intégrale de l'ouvrage inspecté, créant ainsi une cartographie globale de ce dernier.

Lorsque les images sont acquises de façon séquentielle, par exemple depuis un véhicule roulant dans un tunnel, il arrive que deux images successives présentent un recouvrement important, c'est-à-dire que les zones physiques représentées soient d'intersection non vide. Dès lors qu'on travaille dans cette configuration, il peut être intéressant de tenir compte, pour déterminer la nature d'une zone physique donnée, de considérer les prédictions de l'ensemble des images où cette zone est visible. C'est ce travail qui a été réalisé par Schmugge *et al.* [114] : pour deux images successives, le décalage est calculé afin que la prédiction du réseau de l'image en cours puisse être replacé dans l'image précédente. Une fois l'ensemble des prédictions agrégées pour chaque zone physique, le résultat est obtenu par la moyenne des prédictions.

À l'aide d'images RBG-D (*i.e.* des images RGB avec un quatrième canal représentant la distance entre les objets et le capteur), Bang *et al.* [115] proposent une chaîne de traitement complète pour la détection de plusieurs types d'anomalies : fissures, fers apparents et pertes de matières. Les composantes RGB sont utilisées pour la détection et la localisation des anomalies, par un apprentissage d'un modèle Faster R-CNN [116], et la composante de profondeur permet de déduire, dans un second temps, certaines caractéristiques physiques des anomalies détectées. Par exemple, cette méthode réalise une estimation de

la longueur et de la largeur des fissures, de même que de la surface des fers apparents et des pertes de matières. Même si les auteurs n'établissent pas de cartographie globale du tunnel, un recollement entre prédictions est réalisé, ce qui peut représenter une première étape dans la construction d'une telle cartographie. De plus, notons que, dans ces travaux [115], la profondeur n'est utilisée ni pour la reconnaissance des anomalies, ni pour le recollement des prédictions. Une autre méthode couplant la segmentation sémantique avec une reconstruction 3D de l'ouvrage inspecté est présentée dans [117], où un algorithme de SLAM (*Simultaneous Localization And Mapping*) est utilisé conjointement avec une architecture de segmentation U-Net sur des images RGB-D.

Bien que nous n'ayons pas choisi d'orienter les travaux de thèse dans cette direction, ces approches en constituent une perspective intéressante sur le plan opérationnel.

4.2.2 Données 3D

Comparatif d'approches pour la segmentation des fissures Barisin *et al.* [118] réalisent un comparatif de performances entre différents modèles sur des données 3D au sein desquelles sont présentes des fissures de différentes largeurs. Parmi les modèles évalués figurent des modèles sans apprentissage (approches morphologiques, méthodes de percolation, stratégie de chemins minimaux, etc.), un modèle appris non neuronal (forêt aléatoire) et un réseau convolutif (U-Net 3D [119]). Il en ressort, aussi bien en précision qu'en rappel, que les méthodes sans apprentissage présentent une grande variance et se positionnent, en moyenne, en deçà de la forêt aléatoire et du réseau convolutif. C'est ce dernier modèle qui arrive le plus souvent en tête du classement, même si l'écart demeure faible avec la forêt aléatoire (de l'ordre de quelques pourcents).

Notons toutefois que cette étude a été réalisée sur une base de données qualifiées de « semi-synthétiques » par les auteurs. Plus précisément, chaque exemple d'apprentissage et d'évaluation est un volume de dimension 256^3. Pour générer une telle donnée, une image de dimension 256^2 est extraite d'un jeu de données réelles et est répétée selon l'un des axes du volume. La vérité terrain associée à ce volume nouvellement créé est générée de façon procédurale, directement dans $[\![0; 255]\!]^3$, et est ensuite appliquée sur l'ensemble des coupes de ce dernier : chaque voxel correspondant à une fissure est alors remplacé par un voxel noir. On peut, en particulier, souligner l'absence d'éléments perturbateurs (joints, équipements, etc.). Ainsi, il n'est pas garanti que ces données semi-synthétiques rendent fidèlement compte des performances qu'auraient les mêmes modèles évalués sur des données réelles.

Malgré cette réserve, les résultats de Barisin *et al.* [118] tendent donc à montrer que les réseaux de neurones constituent la famille de modèle la plus adaptée pour la segmentation sémantique des fissures.

Fusion de modalités Les données 3D étant constituées de composantes de natures différentes (intensité, profondeur, etc.), la question de la façon dont

sont intégrées ces composantes au sein d'un modèle statistique est une problématique importante.

Dans le cadre de la reconnaissance de fissures par apprentissage profond, Zhou *et al.* [120] proposent différentes stratégies de fusion de modalités, toutes appartenant à la catégorie *early fusion* (*cf.* chapitre précédent). Quatre configurations sont testées :

- Intensité seule ;
- Profondeur brute seule ;
- Profondeur « filtrée » (correction de la courbure globale et réduction du bruit) ;
- Intensité et profondeur brute.

Les auteurs arrivent à la conclusion que la dernière configuration donne les meilleurs résultats, appuyant ainsi l'affirmation que les deux canaux d'informations sont essentiels pour la reconnaissance de fissures.

Rectification de la profondeur Les travaux de Zhang *et al.* [121] portent sur la segmentation de fissures sur chaussées. L'originalité de leur étude est qu'un prétraitement de la profondeur est proposé pour aplanir l'ensemble des profils. Il s'agit, pour chaque point de coordonnées (x, y), dont la profondeur brute est notée $g(x, y)$, de retrancher à $g(x, y)$ la profondeur moyenne sur son voisinage 25×25. Formellement, la profondeur corrigée $\tilde{g}(x, y)$ vérifie alors

$$\tilde{g}(x, y) = g(x, y) - \frac{1}{25 \times 25} \sum_{i=-12}^{12} \sum_{j=-12}^{12} g(i, j) \tag{4.1}$$

Une des limites de cette approche est que la moyenne est sensible aux valeurs extrêmes. Ainsi, cette idée de rectification de la profondeur est prolongée dans [10], article issu de la même équipe de recherche. La nouvelle méthode proposée est articulée en quatre temps :

1. Application de filtres médian 9×9, 11×11, 13×13, 15×15 ;
2. Retranchement du résultat du filtre médian aux profils bruts, et ce, pour chaque filtre ;
3. Centrage et réduction en estimant moyenne et variance sur l'image entière ;
4. Concaténation des représentations produites (*early fusion*).

La figure 4.3 illustre ce procédé.

Bien que le prétraitement de la profondeur soit une thématique récurrente dans de nombreux travaux utilisant des relevés laser, Zhou *et al.* [122] ont montré, dans le cadre de la reconnaissance des fissures sur chaussée, que les réseaux de neurones peuvent être entraînés directement à partir de la profondeur brute si les variations qu'elle présente demeurent faibles (faible courbure générale, absence de rainures et de nids de poules, ornières modérées, etc.). Si cette hypothèse d'homogénéité se trouve souvent vérifiée pour les chaussées, elle ne peut

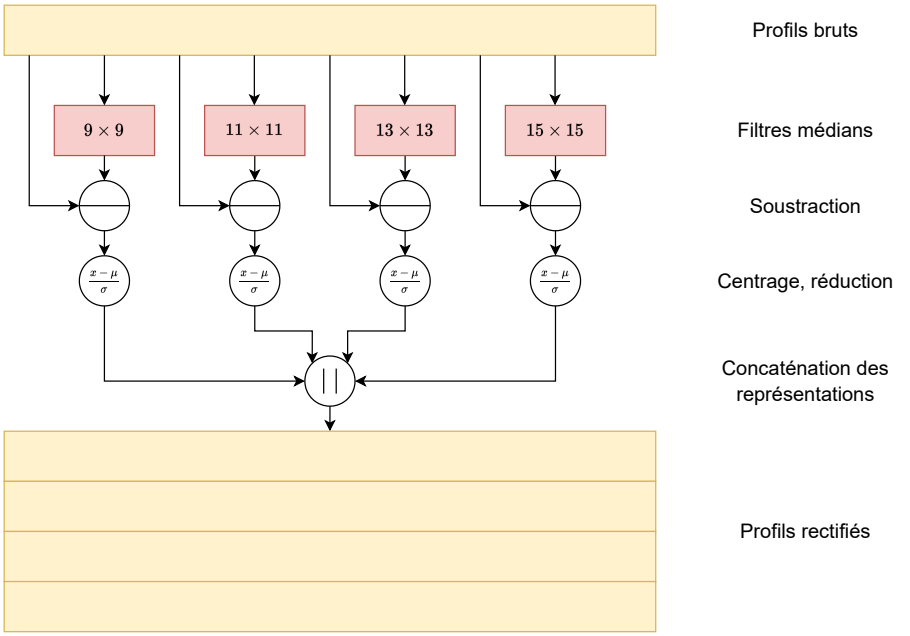


FIGURE 4.3 – Illustration de la méthode de rectification des profils développée dans [10]

en revanche pas être formulée pour les revêtements des tunnels, qui admettent une surface plus complexe et qui comportent, de plus, des équipements.

Même s'il ne s'agit pas à proprement parler d'une méthode d'apprentissage, une approche réalisant le relevé des fissures au pixel près dans des images a été évaluée par Chen *et al.* [123]. L'article présente une chaîne de traitements complète permettant de convertir un revêtement acquis sous forme de nuage de points à une segmentation sémantique des fissures au sein d'un plan inféré à partir de ce nuage. L'étape de segmentation en tant que telle consiste en un seuillage d'Otsu [124] sur les valeurs de profondeur par rapport au plan ainsi considéré. Si cette méthode peut constituer une première approche rapide à mettre en œuvre, elle demeure inadaptée pour des données aussi complexes que les nôtres, pour lesquelles les joints seraient systématiquement prédits comme fissures. Elle reste cependant envisageable pour simplifier le travail d'annotation. Soulignons, par ailleurs, que ce type de méthodes est employé en routine dans les produits associés au LCMS, et distribués par Pavemetrics, pour l'inspection des chaussées.

4.2.3 Positionnement de nos travaux

Bien que la littérature portant sur la détection des anomalies par apprentissage sur les structures de génie civil soit fournie, l'essentiel des travaux réalisés sont dédiés à la reconnaissance de fissures sur chaussée. Or, les contraintes

physiques s'exerçant sur un enrobé ne sont pas les mêmes que sur un parement de tunnel. Il s'ensuit que les fissures auront des caractéristiques et des apparences différentes. De plus, les revêtements sains des tunnels admettent une plus grande variabilité visuelle que les revêtements sains sur chaussée. Ainsi, nombre de méthodes parmi celles présentées dans cette section ciblent des problématiques secondaires pour le cas des tunnels (prise en compte de la topologie des fissures, incitation à détecter des contours nets, etc.). Inversement, certains facteurs de variabilité ne sont pas, ou peu, considérés dans ces mêmes méthodes (courbure de la paroi, équipements, joints, etc.).

Certaines approches, sont, en revanche, particulièrement indiquées dans notre contexte, même si nous n'avons pas eu l'occasion de toutes les expérimenter. L'ensemble des travaux présentés dans cette sous-section est développé dans le chapitre suivant, portant sur l'apprentissage intra-domaine.

Images photographiques Pour les images photographiques, nous mettons en œuvre deux approches de cartographie :

— La première, par quadrillage régulier, porte sur une banque de données dédiée à la classification de quatre types d'anomalie (fissures, fers apparents, zones humides, nids de cailloux), regroupées au sein d'une même classe d'intérêt. Cette base, antérieure à la thèse, avait pour vocation de mener des expérimentations pilotes afin d'évaluer le potentiel des méthodes d'apprentissage automatique. Ces expérimentations étant exploratoires, la base a volontairement été construite de sorte à en réduire la difficulté (taille réduite, classes du jeu d'apprentissage équilibrées). Compte tenu de ces propriétés, nous avons choisi d'utiliser des architectures légères (LeNet [70] et VGG [18]).

— La seconde, par segmentation sémantique, a été appliqué à plusieurs banques de données et vise à reconnaître les fers apparents. À cette fin, nous avons opté pour une architecture SegNet [6] et adopté une stratégie d'annotation visant à regrouper les armatures des fers apparents ainsi que les pertes de matières les composant au sein d'une unique classe d'anomalie, à l'image des travaux de Savino *et al.* [109].

Données 3D Concernant les données 3D, nous disposons d'une seule source annotée de données acquises avec le capteur LCMS. Nous évaluons, à travers une analyse différentielle, l'influence des différentes combinaisons de modalités sur les performances des modèles. De même, de nombreux auteurs soulignant l'importance de la rectification de la profondeur, nous adjoignons à cette étude l'influence de deux modélisations de la profondeur. L'ensemble de ces comparaisons est réalisé pour toutes les expérimentations mises en œuvre sur cette banque de données. Deux approches ont été implémentées :

— Une approche par classification, dans laquelle nous évaluons une stratégie par quadrillage régulier (telle que définie dans le chapitre 3) ainsi que la méthode de Cha *et al.* [9], ayant recours à deux quadrillages et décrite dans ce chapitre.

— Une approche par segmentation sémantique, au sein de laquelle nous employons, à l'instar de la cartographie par segmentation sémantique sur images photographiques, une architecture SegNet [6].

4.3 Apprentissage inter-domaines

Les méthodes que nous regroupons sous le terme d'apprentissage inter-domaines comprennent les méthodes d'adaptation de domaine, les contributions mesurant l'influence du biais de domaine sur les performances des modèles ainsi que les approches qui, même si elles ne traitent pas expressément de l'adaptation de domaine, pourraient être employées à cet effet.

Les méthodes d'adaptation de domaine visent à améliorer les performances d'un modèle évalué sur un jeu de données significativement différent des données d'apprentissage. Dans notre cas, cela correspond, par exemple, à la situation où un modèle appris sur un tunnel est évalué sur un autre, qui lui est inconnu. Dans ce contexte, le jeu d'apprentissage est appelé **domaine source** et le jeu de test **domaine cible**. Une première approche, lorsqu'un modèle généralise mal sur le domaine cible, consiste à labelliser une partie des images de ce domaine pour réitérer le processus d'apprentissage sur l'ensemble des données annotées. Néanmoins, cette stratégie est particulièrement contraignante, puisqu'elle implique un travail d'annotation conséquent. Les stratégies d'adaptation de domaine cherchent ainsi à alléger ce travail d'annotation, voire à éviter intégralement l'étape de labellisation.

À notre connaissance, une seule méthode d'adaptation de domaine a été testée dans notre champ applicatif [125] (les auteurs indiquent, de même, ne pas avoir connaissance de méthodes analogues). Nous la détaillons en section 4.3.3.

Même si elle n'est pratiquement pas traitée en tant que telle pour la cartographie des anomalies, la problématique de variabilité inter-domaines est évoquée dans certaines contributions et son influence sur les performances des modèles est parfois mesurée. Nous signalons également deux familles d'approches qui ont été proposées dans une autre optique que celle de l'adaptation de domaine mais qui pourraient toutefois y être appliquées : l'apprentissage faiblement supervisé et l'apprentissage semi-supervisé.

L'apprentissage faiblement supervisé [126] consiste à utiliser des annotations approximatives (et donc plus rapides à réaliser) tout en tenant compte de cette imprécision tandis que pour l'apprentissage semi-supervisé [127], seule une partie des données du jeu d'apprentissage est labellisée, le restant des exemples, bien que non annoté, demeure utilisé pour l'ajustement des paramètres du modèle. Ces deux stratégies permettent de réaliser un apprentissage conjoint sur les deux domaines. Dès lors, on peut espérer que le biais entre ces domaines aura une incidence moindre sur les performances du modèle appris lorsque ce dernier sera évalué sur le domaine cible. Ces deux approches sont résumées et comparées à l'apprentissage supervisé dans la figure 4.4.

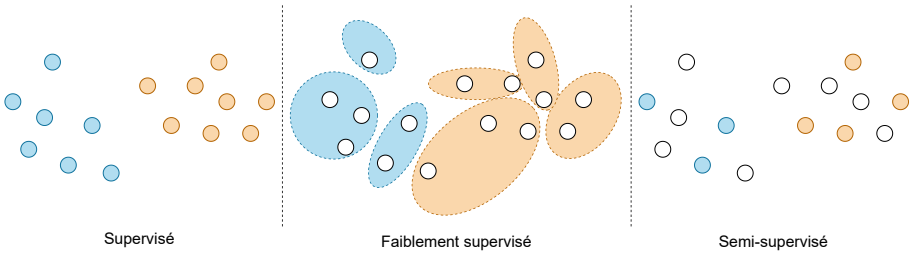


FIGURE 4.4 – Comparaison des différents types de supervision présentées dans cet état de l'art. Chaque point représente un exemple (correspondant à un pixel dans le cas de la segmentation sémantique) et la couleur indique la vérité terrain (bleu : sain ; orange : anomalie ; blanc : non annoté). Apprentissage supervisé : tous les pixels sont annotés ; apprentissage faiblement supervisé : le plus souvent, les pixels sont groupés par lot (par exemple, par sous-images) et le lot reçoit une annotation (contient une anomalie/ne contient pas d'anomalie) ; apprentissage semi-supervisé : seule une partie des pixels est annotée.

4.3.1 Apprentissage actif

Bien qu'il ne traite pas directement de la problématique de la variabilité inter-domaines, un des premiers articles à explorer les limites des approches supervisées est [128]. Feng *et al.* y dressent le constat que la quantité d'exemples annotés nécessaire est conséquente et qu'il est long (et donc coûteux) de constituer de larges jeux de données dans l'espoir qu'ils atteignent un degré suffisant de représentativité. Ils proposent alors une approche reposant sur l'apprentissage actif [129]. Avec cette technique, le réseau convolutif est d'abord appris à partir d'un jeu annoté. Ensuite, de nouvelles données sont présentées au réseau. Un échantillonnage est alors opéré parmi ces nouveaux exemples : ceux pour lesquelles le réseau est le moins certain de sa prédiction sont alors labellisés par un expert avant de rejoindre le jeu d'apprentissage. La contribution majeure de cet article réside dans cette sélection qui permet de se passer d'annotations pour les exemples considérés comme simples par le modèle appris (qui ne sont donc pas ajoutés au jeu d'apprentissage). Si une telle approche est applicable pour la classification, elle est difficile à mettre en œuvre dans le cadre de la segmentation sémantique. En effet, l'utilisateur serait alors contraint d'annoter de larges quantités de pixels isolés dans des images, ce qui peut s'avérer fastidieux.

4.3.2 Influence de la variabilité inter-domaines

Pour les fissures, quelques travaux ont mesuré la perte relative de performance lorsqu'un modèle entraîné sur un site est testé sur un jeu d'exemples issus d'un autre site [130, 131, 132].

Drouyer [130] réalise une évaluation croisée pour la segmentation sémantique : disposant de quatre sources de données, il entraîne, alternativement,

un modèle sur une des sources pour le tester sur les trois autres. En termes de F_1-score, la perte due au biais de domaine est comprise entre 25% et 50% selon les sites, c'est-à-dire que l'on parvient à conserver entre 75% et 50% du F_1-score obtenu sur le jeu de test du même site. Dans le deuxième article, Hallee *et al.* [131] cherchent à mettre au point un classifieur de fissures sur maçonnerie et considèrent deux domaines. Le premier consiste en des prises de vue d'une maquette de mur en maçonnerie qui a été réalisée en laboratoire. Le mur en taille réduite est alors fissuré à la main par application de contraintes de torsion et de pression. Le second est composé d'images de maçonnerie, avec et sans fissures, téléchargées *via Google Images*. Lorsque le modèle est appris sur les images acquises en laboratoire, la perte de F_1-score est de l'ordre de 20% quand le test porte sur les images agrégées depuis internet. Dans le troisième article, Augustauskas *et al.* [132] montrent que l'enrichissement du jeu d'apprentissage par des exemples issus du domaine cible est un levier plus important que l'amélioration des architectures employées. En d'autres termes, une architecture rudimentaire entrainée sur un jeu enrichi est davantage susceptible de donner de bons résultats qu'une architecture sophistiquée apprise sur le jeu initial uniquement.

4.3.3 Adaptation de domaine

La méthode proposée par Liu *et al.* [125] s'inscrit dans une démarche semi-supervisée avec la reconnaissance des fissures pour application. Néanmoins, à la différence des autres méthodes semi-supervisées que nous présentons par la suite, cette approche vise explicitement à réduire le biais de domaine.

Les auteurs postulent que le biais de domaine, pour leur tâche d'intérêt, réside essentiellement dans les variations d'illumination liées aux conditions d'acquisitions (conditions météorologiques, lumière ambiante, etc.). La méthode d'adaptation de domaine mise en œuvre repose alors sur deux principes : un procédé d'augmentation de données d'une part et une fonction de coût visant à confondre les domaines au sein d'espaces de caractéristiques d'autre part.

L'apprentissage est opéré conjointement sur les domaines source et cible. La méthode d'augmentation de données est employée sur l'intégralité des exemples, quel que soit son domaine d'appartenance. Cette méthode consiste à appliquer une correction gamma d'un facteur aléatoire à chaque exemple avant d'y opérer une égalisation d'histogramme. Ensuite, chaque exemple est projeté dans un espace de caractéristiques grâce à un encodeur. Repris de la méthode MK-MMD (*Multi-Kernel Maximum Mean Discrepancies*) [133], un terme de coût mesurant la distance entre les centroïdes des deux domaines projetés dans l'espace de caractéristiques est alors introduit pour pénaliser la disparité entre ces domaines. Pour accroître l'efficacité de la méthode de confusion, plusieurs espaces de caractéristiques sont considérés : en plus de celui produit en sortie de l'encodeur sont concernés les deux espaces issus respectivement des pénultièmes et antépénultièmes blocs de convolution de ce même encodeur.

Dans l'article, l'encodeur est complété par un module neuronal permettant de générer une cartographie par boîtes englobantes mais la méthode proposée

pourrait être adaptée à n'importe quel autre type de cartographie. À l'issue de ce module, un terme de coût supervisé est calculé pour les exemples munis d'annotation (qu'ils proviennent du domaine source ou du domaine cible).

4.3.4 Apprentissage faiblement supervisé

De façon générale, l'apprentissage faiblement supervisé renvoie à toute configuration d'apprentissage dans laquelle les annotations sont empreintes d'une forme d'imprécision. Bien que ce ne soit pas systématique, cette imprécision se matérialise, le plus souvent, par des annotations définies à l'échelle de l'image (l'image contient une anomalie/l'image n'en contient pas). On parle alors de *Multiple Instance Learning* (*MIL*). De telles annotations présentent l'avantage d'être rapides à obtenir mais induisent, en contrepartie, une asymétrie entre classes qu'il faut prendre en compte lors de l'apprentissage. En effet, une image annotée comme saine indique que tous ses pixels appartiennent à cette même classe, contrairement à une image labellisée comme anomalie, pour laquelle on sait uniquement que certains de ses pixels représentent une anomalie, sans que l'on ne connaisse précisément lesquels. Grâce à la réduction du temps d'annotation permise par l'apprentissage faiblement supervisé, il devient envisageable d'annoter une partie des données du domaine cible de cette manière et de réaliser un apprentissage complet sur l'intégralité du domaine source auquel sont ajoutés les exemples du domaine cible qui ont été labellisés.

Approches en deux étapes Les approches en deux étapes que nous présentons concernent uniquement la reconnaissance des fissures. Elles décomposent le problème de détection des fissures en deux sous-problèmes. Une première phase vise à déterminer la localisation grossière des fissures tandis qu'une seconde consiste à affiner ce résultat.

Certains travaux reposent sur l'utilisation des activations d'un classifieur [134, 135, 136] pour la reconnaissance de fissures. Dans [134], Dong *et al.* entraînent un classifieur à distinguer les images comprenant des fissures de celles ne représentant que des revêtements sains. La prédiction au pixel près est réalisée en calculant les activations du classifieur avec la méthode CAM (*Class Activation Mapping*) [137] et en améliorant le résultat à l'aide d'un CRF (*Conditional Random Field*) dense. La même idée est reprise par Konig *et al.* [135] et Inoue *et al.* [136], qui apportent chacun une modification sur l'étape de raffinement : au lieu d'employer un CRF, les premiers multiplient les activations, pixel à pixel, avec le résultat d'un seuillage d'Otsu appliqué à l'image initiale tandis que les seconds réalisent la même opération, mais avec le négatif de cette même image.

Une approche entièrement neuronale est proposée par Xu *et al.* [11]. L'architecture utilisée est composée de trois sous-réseaux. Le premier dans l'ordre d'application est un encodeur repris de VGG16 [18], qui sert à extraire des caractéristiques de haut niveau. Un second réseau génère ensuite une carte de localisation à partir de ces caractéristiques, c'est-à-dire une carte de caractéristiques composée d'un seul canal dont chaque valeur est comprise entre 0 et 1.

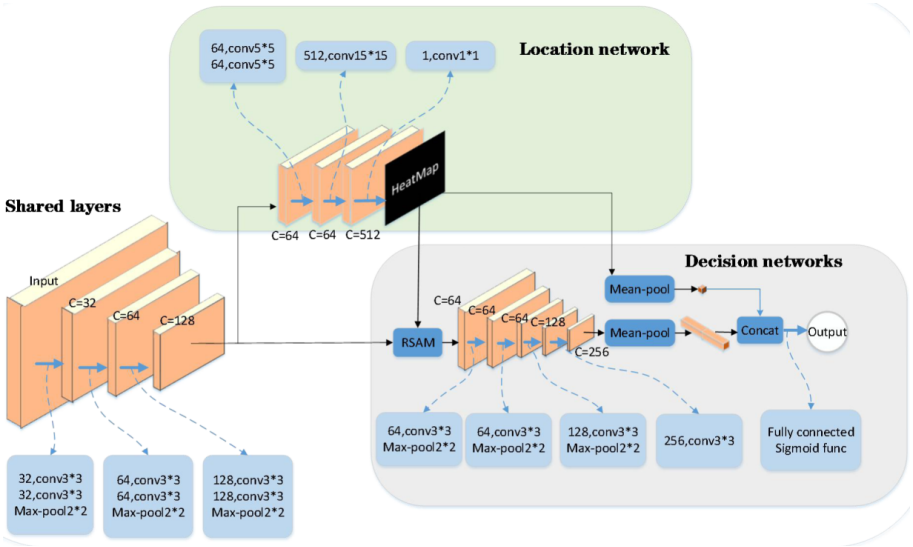


FIGURE 4.5 – Architecture utilisée par l'approche de Xu *et al.* (source : [11])

Un module d'attention vient alors combiner cette carte de localisation avec les caractéristiques produites par l'encodeur pour amplifier ou atténuer certaines zones de l'image. La prédiction est ensuite construite par un autre encodeur. La figure 4.5 donne le détail de cette architecture neuronale.

Si ces approches permettent d'obtenir des relevés de fissures relativement précis, elles requièrent l'apprentissage de classifieurs, et donc la constitution de larges banques de données annotées issues du tunnel à analyser, ce dernier constituant alors simultanément les jeux d'apprentissage et de test pour ce type de méthodes. Elles ne sont donc pas applicables si le nombre d'images considérées est faible, ce qui serait typiquement le cas pour les tunnels courts.

Approches antagonistes Une autre option pour l'apprentissage faiblement supervisé réside dans l'utilisation de réseaux antagonistes génératifs.

Dans [138], ils sont utilisés pour la synthèse de nouvelles données comprenant des anomalies, afin d'enrichir le jeu d'apprentissage. Ces données synthétisées ne requièrent alors aucun travail d'annotation supplémentaire. Bien que cette approche n'ait pas été présentée par ses auteurs comme une approche faiblement supervisée, elle peut toutefois s'en rapprocher puisqu'elle induit la génération d'exemples synthétiques, dont le réalisme n'est pas parfait. La difficulté à obtenir des exemples réalistes constitue la principale limite de cette méthode, puisqu'il faut disposer d'une grande quantité de données réelles pour améliorer la qualité des données synthétiques.

Inspirée de *CycleGAN* [139], une approche impliquant deux paires de réseaux antagonistes génératifs est présentée par Duan *et al.* [12]. Au sein de la première paire, le discriminateur D_1 vise à discerner les vraies images de

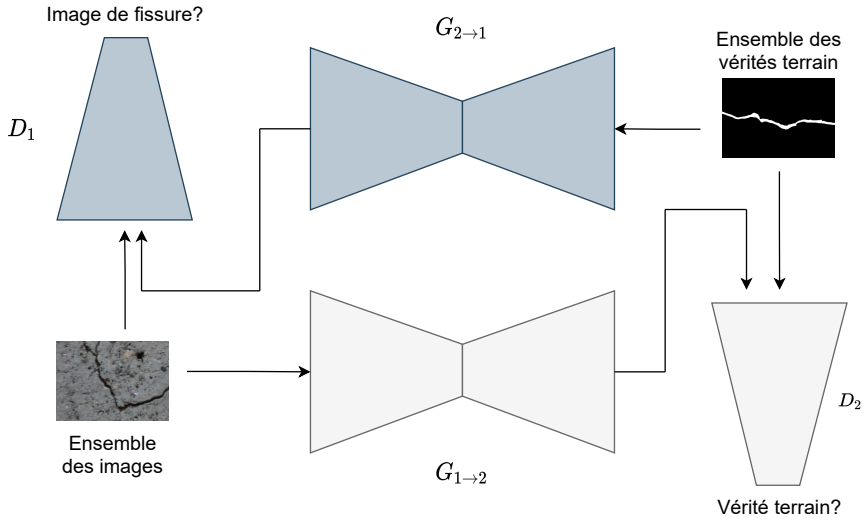
fissures des fausses tandis que le générateur $G_{2\to1}$ est entrainé à produire des images de fissures les plus réalistes possibles. La particularité de ce générateur est qu'il ne prend pas un signal aléatoire pour entrée, mais une vérité terrain de fissure tirée d'un autre jeu de données. Symétriquement, le discriminateur D_2 doit apprendre à reconnaître des vérités terrain de fissures et $G_{1\to2}$ doit apprendre à synthétiser une vérité terrain à partir d'une image. C'est ce dernier générateur, $G_{1\to2}$, qui constitue le modèle de segmentation : il admet une image contenant une fissure en entrée et retourne la vérité terrain associée. La clé de cette méthode, qui garantit la correction de cette association, est la cohérence cyclique introduite à travers un terme de coût faisant tendre $G_{1\to2} \circ G_{2\to1}$ vers la fonction identité. La figure 4.6a résume le fonctionnement de cette approche. Cette idée est poussée plus loin par Zhang *et al.* [13] qui adjoignent aux vérités terrain de fissures des vérités terrains issues de banques de données publiques traitant d'autres problématiques telles que la reconnaissance de contours. Bien que présentées par leurs auteurs comme non supervisées, ces méthodes peuvent être employées dans une optique d'adaptation de domaine. En effet, il est possible de prendre pour base d'images (sans annotation) la réunion des domaines source et cible et de considérer, pour l'ensemble des vérités terrain, les vérités terrain du domaine source. Une telle méthode peut être vue comme faiblement supervisée du fait de l'absence d'appariement entre les images et les vérités terrain du domaine source.

Même si ces approches peuvent paraître élégantes, elles demandent une grande quantité de ressources matérielles (mémoire et capacité de calcul) pour être mises en œuvre. De plus, les quelques exemples de fissures générées par $G_{2\to1}$ et présentés dans [13] (voir figure 4.6b) sont rudimentaires, dans le sens où les revêtements sains sont homogènes et de couleur grise tandis que les fissures se résument à des pixels sombres. Qui plus est, de nombreux artefacts sont visibles. Cette simplicité dans les exemples générés interroge alors sur la capacité des réseaux à modéliser finement ce qui caractérise une fissure.

4.3.5 Apprentissage semi-supervisé

L'apprentissage semi-supervisé peut être mis au service de l'adaptation de domaine en labellisant une partie des données du domaine cible, l'apprentissage portant alors sur le domaine source, entièrement annoté, et sur le domaine cible, qui ne l'est que partiellement.

Approches antagonistes Les réseaux antagonistes génératifs constituent une première voie pour l'apprentissage semi-supervisé. De façon indépendante, Li *et al.* [15] et Shim *et al.* [16] ont proposé deux méthodes de segmentation sémantique dont la mécanique principale est tirée des travaux de Hung *et al.* [14]. Le générateur de la paire de réseaux antagonistes génératifs est une architecture de segmentation dont l'objectif est de localiser les fissures présentes dans les images. Le discriminateur, quant à lui, cherche à distinguer les prédictions issues du générateur de vérités terrain en statuant sur le caractère réaliste de



(a) Fonctionnement de la paire de réseaux antagonistes génératifs

(b) Exemples d'images de fissures générées par $G_{2\to1}$ (source : [13])

FIGURE 4.6 – Illustration de l'approche présentée dans [12, 13]

ces prédictions. Le résultat du discriminateur est alors une carte de caractéristiques, que nous appelons **carte de confiance** par la suite, et qui indique la vraisemblance de la prédiction du générateur en chacun de ses pixels. Le protocole d'apprentissage diffère selon que les exemples présentés aux réseaux sont annotés ou non :

— Pour les exemples annotés, le générateur est appris de façon supervisée avec l'entropie croisée $\mathcal{L}_{ce}$ comme fonction de coût. Deux coûts antagonistes, $\mathcal{L}_{adv}$ et $\mathcal{L}_D$, sont également introduits. Le coût $\mathcal{L}_{adv}$ incite le générateur à leurrer le discriminateur et $\mathcal{L}_D$ pousse ce dernier à discerner les vérités terrain des prédictions du générateur.

— Pour les exemples non annotés, l'idée est la suivante : si une prédiction issue du générateur est considérée comme suffisamment réaliste par le discriminateur, elle peut être employée comme telle et, ainsi, remplacer l'annotation manquante. Pour chaque pixel où la carte de confiance est supérieure à un certain seuil fixé en amont, une entropie croisée $\mathcal{L}_{semi}$ est alors calculée entre la prédiction du générateur et cette même prédic-

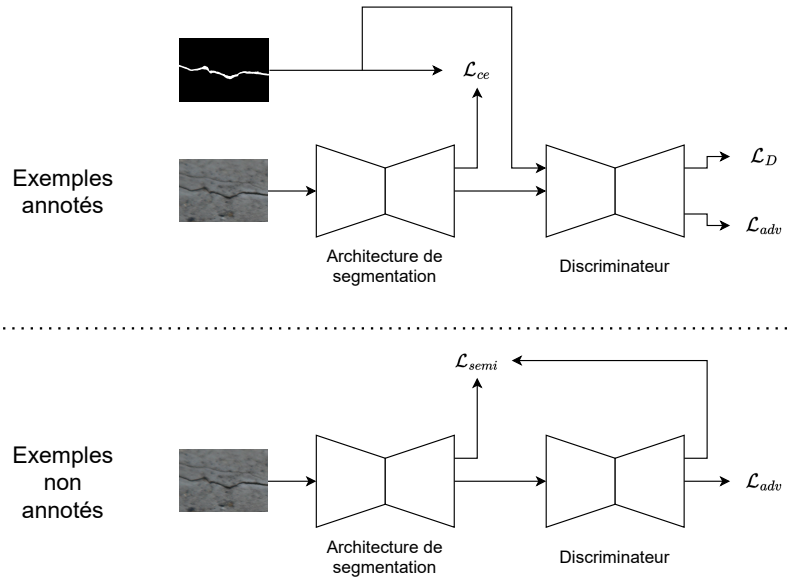


FIGURE 4.7 – Illustration de la méthode de segmentation sémantique semi-supervisée proposée dans [14] et utilisée dans [15, 16]. Le chemin emprunté par les données, ainsi que les coûts associés, varient selon qu'elles soient annotées ou non (voir texte).

tion convertie en vérité terrain. Ce faisant, on permet au générateur de confirmer certaines de ses prédictions sur les données non annotées. Si le coût antagoniste $\mathcal{L}_{adv}$ est encore appliqué au générateur, ce n'est plus le cas pour $\mathcal{L}_D$, qui nécessite la vérité terrain et n'est donc pas utilisable pour les exemples non annotés.

Les grandes lignes de cette méthode sont illustrées en figure 4.7. Bien que cette méthode soit concernée par les problématiques inhérentes à l'apprentissage de réseaux antagonistes génératifs, sa mise en œuvre fructueuse par deux équipes indépendantes laisse penser qu'elle constitue une approche facilement transposable à notre contexte applicatif.

Une autre méthode, reposant également sur les réseaux antagonistes génératifs, est développée par Liu *et al.* [17] (voir figure 4.8). Le générateur apprend à générer des images contenant des fissures à partir d'un signal aléatoire mais le discriminateur a, pour sa part, un double rôle. En plus de déterminer si l'image présentée est réelle ou générée, il doit également attribuer une classe à cette image s'il a prédit que cette dernière était réelle. L'intérêt de cette approche réside dans l'architecture du discriminateur, qui « mutualise » (dans un sens qui nous détaillons par après), les neurones de la dernière couche pour les deux tâches de classification (classification parmi un certain nombre de classes d'intérêt d'une part, et classification « exemple réel/généré » d'autre part). Cette mutualisation contraint le réseau à « prédire » une classe pour les exemples non

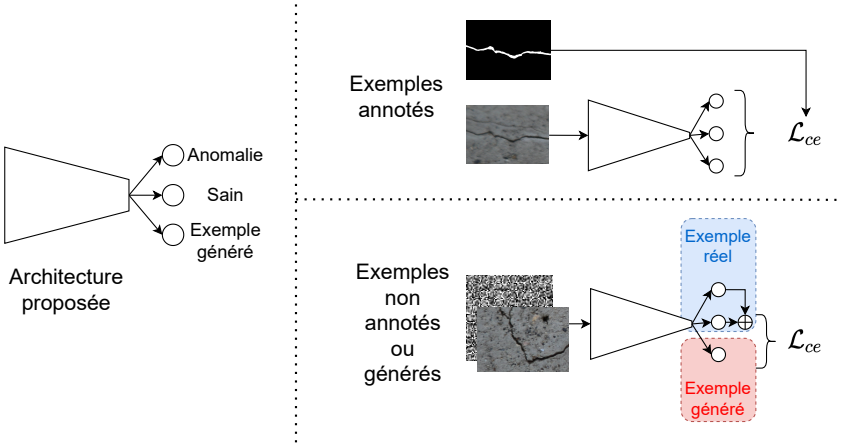


FIGURE 4.8 – Illustration de la méthode de classification semi-supervisée proposée dans [17], présentée ici avec deux classes d'intérêt (voir texte). La partie située à gauche correspond au discriminateur de la paire de réseaux antagonistes génératifs. La partie de droite présente l'apprentissage de ce discriminateur selon la nature des exemples (annotés, non annotés ou générés).

annotés lors de l'apprentissage du modèle. Formellement, si la tâche considérée vise à classifier une image parmi K classes, le discriminateur est alors un classifieur à $K+1$ classes : les K classes d'intérêt auxquelles on ajoute une classe pour les images générées. Le discriminateur modélise alors une fonction

$$f : \mathcal{I}_{W,H,D} \to [0;1]^{K+1} \tag{4.2}$$

où la classe $K+1$ représente la classe des exemples produits par le générateur. Pour les données annotées, le discriminateur est appris de façon supervisée avec l'entropie croisée comme fonction de coût. Pour les données non annotées, le coût introduit est également l'entropie croisée mais il ne porte pas directement sur f, mais sur

$$\begin{aligned} \tilde{f} : \mathcal{I}_{W,H,D} &\to [0;1]^2 \\ (i,j,k) &\mapsto \begin{pmatrix} \sum_{n=1}^{K} f(i,j,k)_n \\ f(i,j,k)_{K+1} \end{pmatrix} \end{aligned} \tag{4.3}$$

où $f(i,j,k)_n$ est la n-ième composante du vecteur $f(i,j,k)$. En procédant de la sorte, on peut utiliser les images non annotées en optimisant le fait qu'elles ne soient pas classées comme générées. En remontant à la prédiction donnée par f, on obtient alors la classe prédite pour ces images.

Approches par distillation de connaissances Wang *et al.* [140] adaptent, dans le cadre de la segmentation sémantique des fissures, la méthode semi-supervisée introduite par Tarvainen *et al.* [141]. Cette dernière approche pro-

pose de considérer deux réseaux de neurones ayant une architecture commune mais des valeurs de paramètres différents.

— Pour les exemples annotés, l'un des deux réseaux, appelé **réseau étudiant**, est appris de façon supervisée. Les paramètres de l'autre réseau, appelé **réseau enseignant**, demeurent alors inchangés.

— Pour les exemples non annotés, la méthode cherche à faire coïncider les prédictions des deux réseaux de neurones en introduisant un coût mesurant l'écart entre elles. Naturellement, pour que ce coût représente un intérêt lors du processus d'apprentissage, il faut que les paramètres des deux réseaux soient distincts. Pour garantir que cela soit le cas, le procédé de mise à jour des paramètres diffère selon le réseau considéré. Pour le réseau étudiant, c'est la rétropropagation du gradient qui est utilisée. Quant au réseau enseignant, ses poids sont uniquement mis à jour par moyenne pondérée exponentielle : pour chaque paramètre w_2 de ce second réseau, sa valeur après mise à jour vaut

$$\alpha w_2 + (1 - \alpha) w_1 \tag{4.4}$$

où w_1 est le paramètre correspondant dans le premier modèle et α un facteur d'inertie. En raison de cette inertie, le réseau enseignant est moins sujet aux fluctuations de la descente de gradient. Ainsi, c'est ce dernier réseau qui est conservé pour l'évaluation.

En comparaison des approches antagonistes, cette méthode ne souffre pas de problèmes d'instabilité lors de l'apprentissage. Elle requiert toutefois de stocker en mémoire deux réseaux d'architectures identiques.

4.3.6 Positionnement de nos travaux

Les stratégies dadaptation de domaine sont encore peu étudiées pour répondre à la problématique du biais de domaine dans notre contexte opérationnel, alors même que de telles méthodes ne sont pas rares en dehors de notre champ applicatif. L'approche développée par Liu *et al.* [125] représente, à notre connaissance, la seule méthode dédiée à la reconnaissance des anomalies et visant à prendre en compte le biais de domaine. Elle est toutefois parue tardivement au regard du calendrier de la thèse (pré-publication en novembre 2021, publication en revue en janvier 2023). Nous n'avons ainsi pas eu l'occasion de la mettre en œuvre.

Les approches semi- et faiblement supervisées peuvent être employées pour annoter partiellement le domaine cible et, ainsi, améliorer les performances du modèle sur ce jeu. Néanmoins, ces approches restent coûteuses en termes de coût mémoire et de puissance de calcul, requérant la plupart du temps l'utilisation conjointe de plusieurs réseaux de neurones. De plus, elles reposent parfois sur des réseaux antagonistes génératifs, qui sont connus pour présenter un apprentissage parfois instable.

Nous avons alors implémenté et évalué deux stratégies. La première approche mise en œuvre est non supervisée et est issue des travaux de Sun *et*

al. [21], qui se rapprochent de ceux de Liu *et al.* [125] présentés dans cet état de l'art, dans le sens où les deux visent à créer un espace de caractéristiques commun aux deux domaines. La seconde méthode expérimentée est faiblement supervisée et a été développée par nos soins. Elle consiste à ajuster un nombre réduit de paramètres stratégiques d'un réseau de neurones appris sur le domaine source avec une fraction annotée du domaine cible. En comparaison des stratégies faiblement et semi-supervisées développées dans ce chapitre, cette approche ne nécessite pas l'emploi de réseaux de neurones additionnels et est donc plus légère à implémenter, que ce soit en termes de coût mémoire ou de temps de calcul. Ces deux approches sont détaillées dans le chapitre 6.

Conclusion du chapitre

Nous avons vu, dans cet état de l'art, qu'il existait une grande quantité de travaux dans notre champ applicatif. Il en ressort les éléments suivants :

- Les fissures représentent les anomalies les plus considérées dans les méthodes de segmentation sémantique pour la cartographie des anomalies sur les constructions de génie civil. En raison de la simplicité de leur apparence et de leur intérêt opérationnel, elles ont fait l'objet de travaux bien avant l'émergence des méthodes d'apprentissage profond. Depuis, ces méthodes y ont été appliquées et ont, de plus, fait l'objet de développements spécifiques, comme la prise en compte de la topologie.
- Peu de travaux ont cherché à cartographier les anomalies présentes à la surface des revêtements de tunnels à l'aide de données LCMS. De telles contributions ont toutefois été proposées à partir d'autres types de données 3D (LIDAR, RGB-D, etc.).
- Pour les données 3D, l'utilisation conjointe des deux canaux d'information (intensité et profondeur) semble être une piste prometteuse pour améliorer les résultats. De plus, différents prétraitements de la profondeur ont été mis en œuvre et évalués. Néanmoins, ces travaux portent le plus souvent sur des relevés de chaussées, dont on sait que la surface est plus simple que celle des tunnels.
- Les études inter-domaines montrent que le biais entre domaines est conséquent. Nous n'avons recensé qu'une seule contribution portant sur l'adaptation de domaine pour la reconnaissance d'anomalies dans notre cadre applicatif.
- D'autres approches, utilisables pour réduire le biais de domaine, ont été proposées dans la littérature, même si elles l'ont été dans un but différent.

Au vu de cet état de l'art, nous positionnons nos travaux de la façon suivante :

- Pour l'apprentissage intra-domaine, nous mettons en œuvre des méthodes d'apprentissage profond pour la cartographie des anomalies sur nos données. Pour ce faire, et compte tenu de la faible quantité de données dont nous disposons, nous nous sommes orientés vers des architectures légères. De plus, nous implémentons et évaluons différentes approches concernant la modélisation de la profondeur ainsi que la fusion des modalités pour les données LCMS.
- Pour l'apprentissage inter-domaines, nous quantifions le biais de domaine à travers une évaluation croisée entre différents ouvrages.

Nous évaluons alors deux méthodes d'adaptation de domaine : une première approche non supervisée, inspirée des travaux de Sun *et al.* [21], et une seconde, faiblement supervisée, conçue et mise en œuvre par nos soins.

Chapitre 5

Apprentissage intra-domaine pour la détection d'anomalies au sein de structures de génie civil

Dans ce chapitre, nous présentons les différentes expérimentations qui ont été réalisées pour la cartographie des anomalies. Comme nous ne faisons pas d'évaluation croisée entre les différents sites dont sont issues les données (cette étude multi-sites est l'objet du chapitre 6), nous ne nous intéressons qu'à la capacité des réseaux de neurones à modéliser la variabilité au sein d'un même ouvrage. Nous explorons deux méthodes de cartographie, une par quadrillage régulier et l'autre par segmentation sémantique, et ce aussi bien pour les images photographiques que pour les relevés laser. Pour ce dernier type de données, nous évaluons, de plus, l'incrément apporté par l'information de profondeur. Pour toutes les méthodes présentées, nous nous inscrivons dans un cadre binaire, c'est-à-dire que seules deux classes sont considérées : une première classe pour les revêtements sains et une seconde pour les anomalies. Selon le contexte, la classe d'anomalies comprendra un unique type d'anomalie ou, à l'inverse, en englobera plusieurs.

Sommaire

5.1 Images photographiques

5.1.1 Cartographie par quadrillage régulier

Résumé du protocole expérimental

- **Architectures** LeNet, VGG
- **Poids initiaux** Aléatoires
- **Époques** 250 • **Optimiseur** *SGD*
- **Augmentation de données** Aucune
- **Critère d'optimisation** Entropie croisée pondérée
- **Critère de sélection du modèle à évaluer**
  – Modèle issu de la dernière époque
- **Objectifs**
  – Évaluer le potentiel des réseaux de neurones dans notre champ applicatif
  – Comparer les performances de ces modèles avec celles des forêts aléatoires sur un jeu de taille réduite

5.1.1.1 Méthodologie

Cette expérimentation s'inscrit dans la continuité de travaux déjà réalisés au sein du groupe ENDSUM [32, 33]. Ces derniers portent sur l'étude d'une méthode de cartographie par quadrillage régulier des anomalies, méthode testée, entre autres, sur le tunnel de Rive-de-Gier.

Les jeux utilisés correspondent à la « base pour la classification » que nous avons présentée au chapitre 2, section 2.1.6, et dont la composition est rappelée dans le tableau 5.1. Chaque exemple de la base représente alors une sous-image de dimension 101×101 pixels qui a été extraite d'une image en pleine résolution (de résolution 1912×1081 pixels) puis annotée. Cette base est, ainsi, exclusivement composée de sous-images et ne couvre donc qu'une faible portion des images captées dans le tunnel, de l'ordre de 0,45%. Quatre types d'anomalies sont considérées : les fers apparents, les fissures, les nids de cailloux et les zones humides. En raison du faible nombre de représentants pour chacune de ces anomalies, ces différentes classes sont regroupées en une seule pour réduire le problème de classification au cas binaire (anomalie/revêtement sain). Ainsi, un modèle appris sur cette base peut détecter la présence d'une anomalie dans une image, mais pas en déterminer la nature. Pour établir la localisation des anomalies, la stratégie de cartographie par quadrillage régulier décrite au chapitre 2, section 3.1.3, est appliquée.

Dans la précédente étude [32, 33], un classifieur reposant sur une forêt aléatoire [34] a été mis en œuvre. Une forêt aléatoire est un modèle constitué d'un ensemble d'arbres de décision, deux à deux distincts, et dont la prédiction est obtenue par un vote majoritaire entre les prédictions des arbres. Pour garantir l'indépendance de ces différents votes, chaque arbre n'a accès qu'à une partie

	Apprentissage	Test
Sain	594	77
Anomalie	600	201

TABLEAU 5.1 – Composition (nombre d'exemples) des deux jeux de données

conv5-6
avgpool-2
conv5-16
avgpool-2
conv1-120
fc-84
dropout (0.5)
fc-2
Softmax

(a) Réseau proche de LeNet

Encodeur de VGG19 [18]
fc-32
fc-32
fc-2
Softmax

(b) Réseau proche de VGG

conv$k$**-**m	couche de convolution à m filtres dont le noyau est de dimension $k \times k$
avgpool-k	couche de pooling de fenêtre $k \times k$
fc-n	Perceptron multicouches à n neurones
dropout (p)	dropout avec une probabilité de p

FIGURE 5.1 – Présentation des architectures convolutives (la convention utilisée pour les notations est celle de Simonyan *et al.* [18])

de l'information représentant les exemples. La forêt considérée dans cette étude comprend 200 arbres. Contrairement aux réseaux de neurones, qui opèrent directement sur les niveaux de gris des images à traiter, les forêts aléatoires les considèrent par le biais de caractéristiques extraites de ces images et définies en amont de l'apprentissage. En l'occurrence, les caractéristiques utilisées sont les attributs de textures d'Haralick [142], les histogrammes de gradients orientés (HOG) [91] et le descripteur mCentrist [143].

Notre expérimentation consiste alors à entraîner deux réseaux de neurones sur ces données pour pouvoir comparer les résultats nouvellement obtenus avec ceux des travaux antérieurs. Deux architectures ont été testées : un réseau de faible profondeur, comparable à LeNet [70], et un réseau comportant davantage de couches, proche de VGG19 [18] (*cf.* chapitre 3, section 3.1.2). Dans un souci de concision, nous appellerons « LeNet » le premier modèle et « VGG » le second. La figure 5.1 présente le détail de ces deux architectures. La base étant dépourvue de jeu de validation, nous évaluons, pour chacun des deux réseaux, le modèle obtenu à l'issue du processus d'apprentissage. De manière analogue au travail originellement réalisé sur cette base, aucune technique d'augmentation de données n'a été appliquée.

	E	EP	P	R	F_1
Forêt aléatoire [33]	91.73	91.88	96.84	91.54	94.12
Réseaux de neurones	96.76	96.16	98.00	97.51	97.75

TABLEAU 5.2 – Scores obtenus (en %) sur les jeux de test par les différents modèles pour la classification de sous-images (E : exactitude ; EP : exactitude pondérée ; P : précision ; R : rappel ; F_1 : F_1-score)

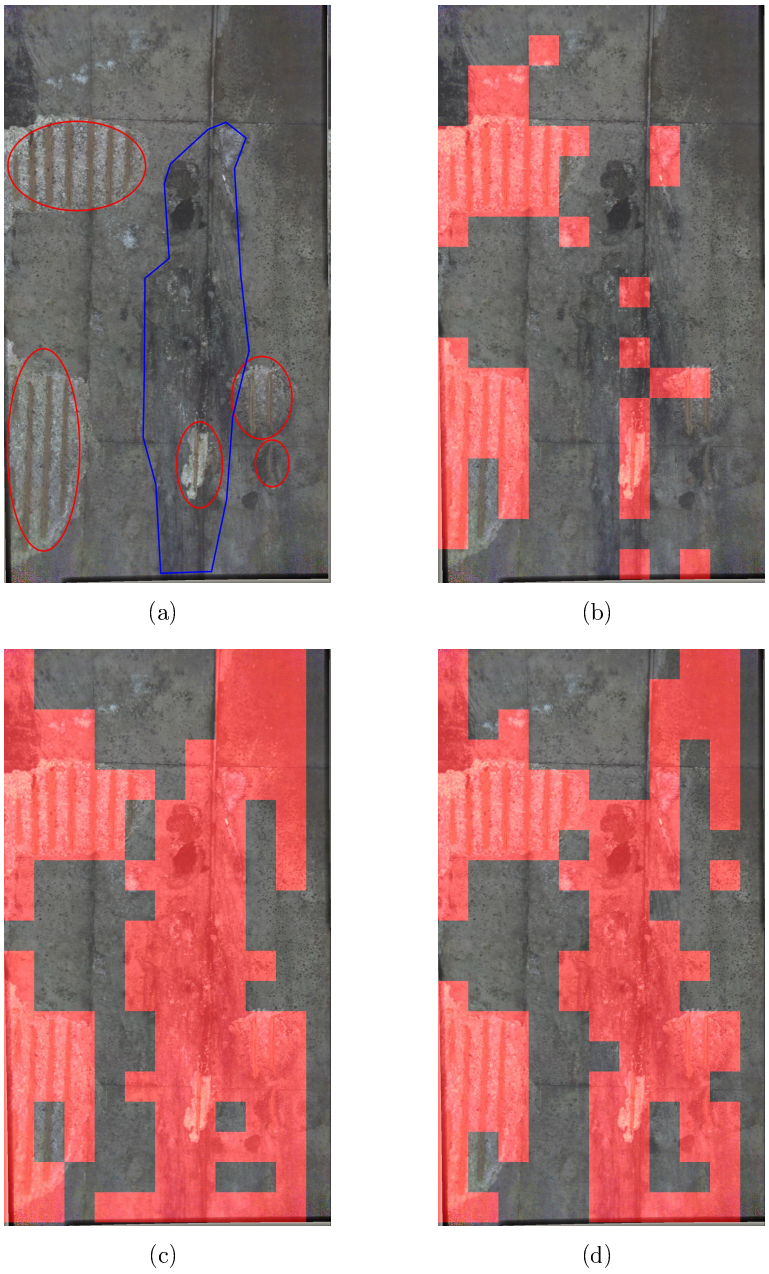
5.1.1.2 Résultats

Les résultats de la classification sont reportés dans le tableau 5.2. Notons que les deux réseaux de neurones réalisent exactement la même performance (*i.e.* même matrice de confusion). On peut y voir que l'ensemble des modèles présentent des scores supérieurs à 90%, toutes métriques confondues.

La figure 5.2 présente le résultat de l'algorithme de fenêtre glissante (sans recouvrement) sur un exemple disjoint du jeu d'apprentissage. On observe que, si la forêt aléatoire réussit à identifier les fers apparents, elle ne parvient en revanche pas à trouver la potentielle zone humide. Les réseaux de neurones parviennent à détecter cette dernière anomalie correctement mais au prix d'un plus grand nombre de faux positifs. Ce phénomène de sur-détection est encore plus manifeste lorsque l'on infère les prédictions des modèles sur des images ne comportant que peu d'anomalies. Dans la figure 5.3, trois exemples sont présentés aux trois modèles : deux représentant exclusivement des revêtements sains et un troisième présentant une possible zone humide. On constate que les résultats obtenus sur ces images ne sont pas conformes aux scores précédemment calculés puisque de nombreuses fausses alarmes sont présentes. Le principal facteur pouvant expliquer cette mauvaise performance est la faible représentativité des exemples des jeux de données. En effet, compte tenu de leur taille, il est vraisemblable que ces derniers ne rendent pas suffisamment compte de la variabilité visuelle du tunnel en question.

Notons que, même si les réseaux de neurones ont de meilleurs résultats sur le jeu de test que la forêt aléatoire, c'est ce dernier modèle qui réalise, au moins visuellement, la meilleure performance sur les trois exemples de la figure 5.3. On peut émettre l'hypothèse que les réseaux de neurones se sont sur-spécialisés sur les données du jeu d'apprentissage là où la forêt aléatoire utilise des cartes de caractéristiques extraites de ces données. Ces cartes ayant été choisies pour leur intérêt dans la tâche à résoudre (les fissures sont caractérisées par un gradient prononcé et détectable dans le HOG, les zones humides et nids de cailloux présentent une granularité élevée que l'on pourrait déceler dans les attributs de texture de Haralick, etc.), la dépendance de la forêt aléatoire aux données d'apprentissage semble donc plus faible que celle des réseaux de neurones. L'ajout d'augmentation de données ou encore l'utilisation d'encodeur pré-appris pourraient probablement compenser en partie ce défaut des réseaux de neurones. Néanmoins, l'enrichissement de la base d'apprentissage paraît indispensable pour obtenir un modèle ayant une réelle utilité opérationnelle.

Cette expérimentation a donné lieu à deux publications, en conférences nationale [144] et internationale [43].



(a) (b)

(c) (d)

FIGURE 5.2 – Cartographie des anomalies par classification sur un exemple de revêtement en béton avec l'algorithme de forêt aléatoire (b), le réseau LeNet (c) et le réseau VGG (d). Dans (a), les fers apparents sont entourés en rouge tandis qu'une possible zone humide est délimitée en bleu. Les anomalies détectées sont en rouge dans les prédictions.

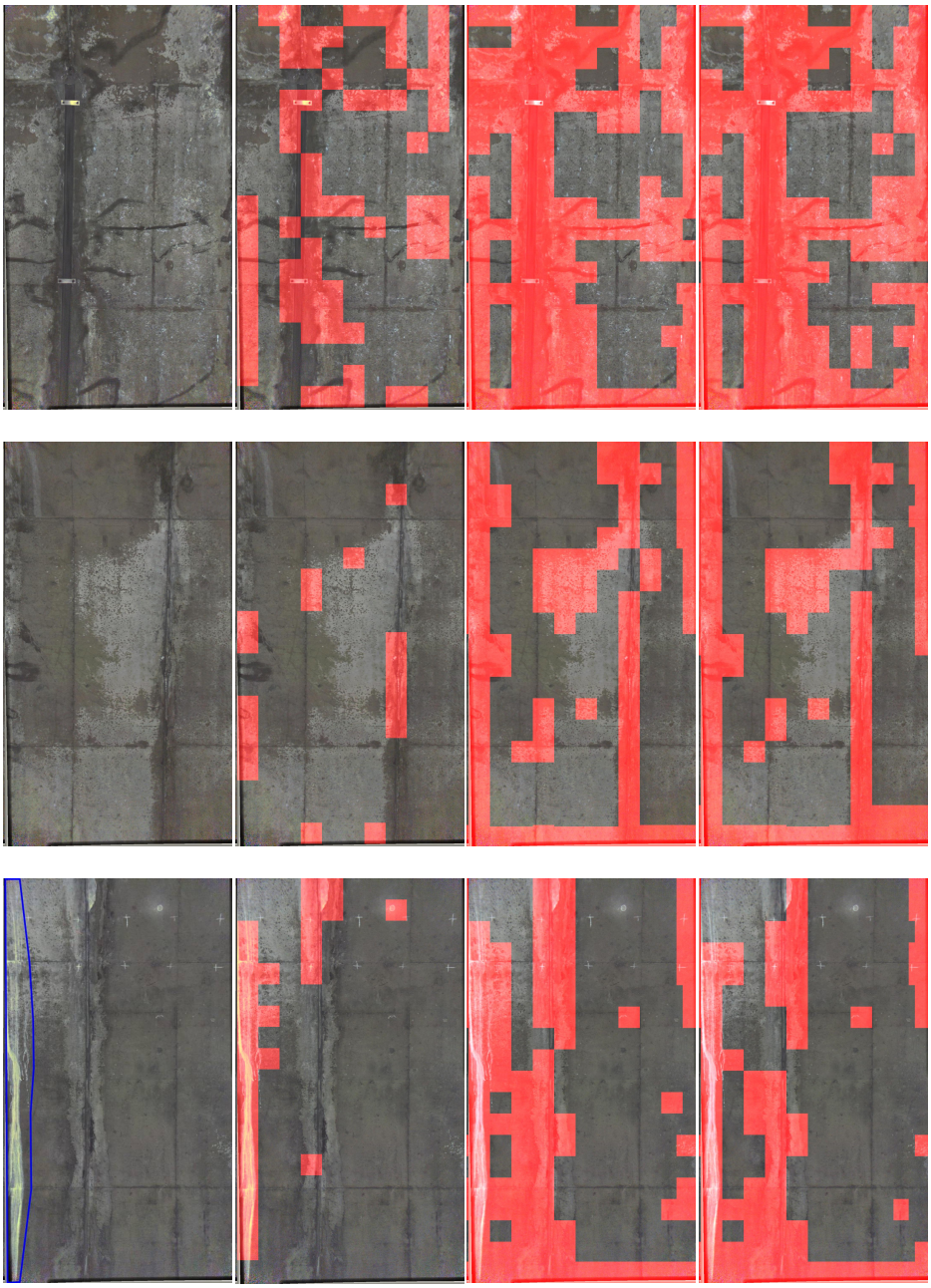


FIGURE 5.3 – Exemples de prédictions pour les trois modèles (Forêt aléatoire, réseaux LeNet et VGG) sur trois images du tunnel de Rive-de-Gier – **1ère colonne :** images à cartographier (une possible zone humide est entourée en bleu au sein de l'exemple de la troisième ligne) ; **2ème colonne :** inférence par la forêt aléatoire ; **3ème colonne :** inférence par le réseau LeNet ; **4ème colonne :** inférence par le réseau VGG.

5.1.2 Cartographie par segmentation sémantique

Résumé du protocole expérimental

- **Architecture** SegNet
- **Poids initiaux** Pré-appris (encodeur), aléatoires (décodeur)
- **Époques** 1000 • **Optimiseur** *Adam*
- **Augmentation de données** Drouyer [130]
- **Critère d'optimisation** Entropie croisée pondérée
- **Critères de sélection des modèles à évaluer**

Tunnels piéton et routier :
– Modèle issu de la dernière époque
Base CODEBRIM :
– Modèle maximisant le F_1-score sur le jeu de validation
– Modèle maximisant l'exactitude pondérée sur ce même jeu

- **Objectifs**

– Évaluer les performances des réseaux de neurones sur des jeux de données de grande taille et en l'absence de biais de domaine
– Évaluer l'intérêt d'une approche multi-échelles pour prendre en compte la variabilité d'échelle des anomalies au sein d'un même site
– Comparer deux critères de sélection du modèle (pour la base CODEBRIM) dans un contexte de déséquilibre modéré entre classes

5.1.2.1 Méthodologie

Dans cette expérimentation, nous mettons en œuvre une architecture de segmentation, SegNet [6] (*cf.* chapitre 3, section 3.1.3), pour la reconnaissance des fers apparents. Ce procédé est réitéré indépendamment pour trois des cinq sources composant la base pour la segmentation sémantique : le tunnel piéton, le tunnel routier de Rive-de-Gier et la base CODEBRIM. Les deux autres sources de données (représentant le bâtiment universitaire) étant dépourvues de jeux d'apprentissage, elles ne sont pas considérées pour cette expérimentation. Pour chaque apprentissage, l'encodeur employé dans l'architecture SegNet est un VGG16 pré-appris sur le jeu de données ImageNet, et dont les poids sont fournis par la bibliothèque PyTorch. Les paramètres du décodeur sont, quant à eux, initialisés aléatoirement selon la méthode de Glorot *et al.* [62] présentée dans le chapitre 3, section 3.1.1.4. Aucun paramètre du modèle n'est figé lors de l'apprentissage.

Pour chaque site, l'apprentissage comprend 1000 époques et les modèles sont optimisés avec Adam [66]. CODEBRIM étant scindé en trois jeux, le jeu de validation sert à sélectionner deux modèles à évaluer parmi ceux générés lors de l'apprentissage : les modèles ayant réalisé la meilleure exactitude pondérée pour l'un, le meilleur F_1-score pour l'autre, sont tous deux évalués sur le jeu

Site	Taille des images	Apprentissage	Validation	Test
Rive-de-Gier	1912×1081	1010	–	418
Rive-de-Gier (restreint)	1912×1081	–	–	27
CODEBRIM	Variable	315	22	29
Tunnel piéton	4896×3672	134	–	78

TABLEAU 5.3 – Composition des jeux des différents sites utilisés pour la cartographie par segmentation sémantique

de test. Nous cherchons ainsi à comparer l'influence de ces deux critères de sélection sur les performances du modèle sélectionné. Concernant les tunnels piéton et routier, pour lesquels aucun jeu de validation n'est disponible, c'est le modèle obtenu à l'issue de la millième époque qui est considéré. Le jeu de test du tunnel de Rive-de-Gier étant majoritairement composé d'images dépourvues de fers apparents, de l'ordre 94%, nous considérons également un jeu de test **restreint**, rassemblant uniquement les 6% d'images présentant un ou plusieurs fers apparents. Cette dernière configuration se rapproche ainsi de la composition de la majeure partie des jeux de données publiquement disponibles dans notre champ applicatif (pour lesquels chaque image présente une anomalie) et nous permet aussi d'évaluer les performances des modèles au voisinage des anomalies. En effet, pour le cas des fers apparents, ces voisinages peuvent présenter des zones humides caractérisées par une texture visuelle granuleuse proche de celle des pertes de matière des fers apparents. Il est alors intéressant d'évaluer plus fidèlement la capacité des modèles à faire la différence entre ces zones humides et les fers apparents. Le tableau 5.3 résume la composition des différents jeux.

L'apprentissage direct sur des images aussi résolues étant techniquement délicat en termes de capacités informatiques, les images des jeux d'apprentissage et de validation sont découpées en sous-images, de façon analogue à ce que nous avons mis en place pour la cartographie par quadrillage régulier. La taille des sous-images est fixée à 256×256 pixels et l'origine du quadrillage employé est située au niveau du coin supérieur gauche des images. Dans le cas où les dimensions ne sont pas des multiples de 256, les bords résiduels droit et bas ne sont pas pris en compte. Notons toutefois qu'un tel découpage peut poser un problème au niveau des vérités terrain, qui ont été relevées sur les images en pleine résolution. En effet, il est possible qu'une sous-image se retrouve intégralement incluse dans la perte de matière d'un fer apparent sans pour autant que les armatures métalliques de ce dernier ne soient visibles, risquant, comme évoqué dans le chapitre 2, section 2.1.6, de rendre les annotations incohérentes.

Nous choisissons alors de sous-échantillonner les images. Appliqué seul, ce procédé se traduirait par une diminution importante du nombre des exemples des jeux d'apprentissage et de validation, puisque la résolution des sous-images extraites demeure de 256×256 pixels. Afin de conserver un maximum d'exemples au sein de ces jeux, nous considérons simultanément un ensemble de facteurs

Site	F	#Sous-images
Tunnel piéton	$\{2^{-2}, 2^{-3}\}$	1876 (1608 + 268)
Tunnel de Rive-de-Gier	$\{2^{-1}\}$	6060
CODEBRIM	$\{2^{-2}\}$	3173

TABLEAU 5.4 – Caractéristiques des jeux d'apprentissage pour les trois sites considérés après extraction des sous-images. Toutes les sous-images admettent une résolution de 256×256 pixels.

de redimensionnement. Pour ce faire, chaque image des jeux d'apprentissage et de validation est dupliquée autant de fois qu'il y a de tels facteurs. Puis, nous extrayons des sous-images dans chacune des images redimensionnées selon ces mêmes facteurs avant de regrouper l'ensemble de ces dernières au sein des jeux d'apprentissage et de validation. Précisons également que cette opération est réalisée indépendamment sur chacun des deux jeux, si bien que les jeux d'apprentissage et de validation restent strictement disjoints. Pour chaque source de données, l'ensemble des facteurs de redimensionnement considérés, noté F, est de la forme

$$F = \{2^{-N} \mid N_{\min} \leq N \leq N_{\max}\} \tag{5.1}$$

où $N_{\max}$ est fixé de telle sorte que le plus petit côté de la plus petite image redimensionnée selon le facteur $2^{-N_{\max}}$ soit supérieur à 256, permettant ainsi d'en extraire une sous-image, et $N_{\min}$ est empiriquement déterminé de sorte à minimiser le risque mentionné auparavant. Notons qu'il n'est pas possible de s'en prémunir de façon certaine par le seul biais de quadrillages réguliers. En effet, on ne peut pas exclure qu'à certains facteurs de redimensionnement, un fer apparent se trouve à cheval sur plusieurs sous-images, sans pour autant que les armatures métalliques figurent dans chacune d'elles. Répondre à cette problématique de manière exacte est techniquement réalisable mais aurait demandé une annotation de ces armatures métalliques. C'est donc un compromis entre, d'un côté, l'effort d'annotation, et, de l'autre, la finesse de la vérité terrain, qui a été fait.

La composition des jeux d'apprentissage pour chacun des sites, ainsi que l'ensemble des facteurs de redimensionnement utilisés, sont détaillés dans le tableau 5.4. Pour le tunnel de Rive-de-Gier et CODEBRIM, un seul facteur de redimensionnement est employé. Pour le tunnel piéton, nous en considérons deux. La figure 5.4 donne un exemple des sous-images extraites d'une image du tunnel piéton, selon ces deux facteurs de redimensionnement.

Lors de la phase d'apprentissage, nous appliquons, pour chaque exemple de chaque époque, la méthode d'augmentation de données décrite par Drouyer [130] et que nous détaillons dans l'algorithme 1. Précisons qu'au moment de la rotation, le contexte entourant la sous-image n'est pas employé, pour des raisons d'implémentation. Ainsi, les pixels présents dans ce voisinage et se trouvant situés, après rotation, au sein de la sous-image résultante se voient attribuer une valeur nulle. Également, après l'étape du redimensionnement, la dimension de la sous-image est ramenée à 256×256 pixels, soit par remplissage

(a) Image initiale (b) Échelle 2^{-2} (c) Échelle 2^{-3}

FIGURE 5.4 – Illustration des tailles relatives des sous-images 256×256 pixels à différents facteurs d'échelle sur le tunnel piéton. L'ensemble des sous-images extraites sur les deux images de droite est inclus dans le jeu d'apprentissage.

Algorithme 1 : Algorithme d'augmentation de données proposé par Drouyer [130] et appliqué sur chaque sous-image 256×256.
$\mathcal{U}$ et $\mathcal{U}_D$ désignent respectivement les lois uniformes continue et discrète.

Entrée : La sous-image à traiter X
$\alpha \sim \mathcal{U}(0.5, 1.5)$;
$\beta \sim \mathcal{U}(-0.15, 0.15)$;
$X \leftarrow \alpha(X - \bar{X}) + \bar{X} + \beta$ (où $\bar{X}$ est la moyenne des valeurs de X) ;

$X \leftarrow$ Miroir_horizontal(X) avec une probabilité de $\frac{1}{2}$;
$X \leftarrow$ Miroir_vertical(X) avec une probabilité de $\frac{1}{2}$;

$\theta \sim \mathcal{U}_D(0, 359)$;
$X \leftarrow$ Rotation(X, θ) ;

$\phi \sim \mathcal{U}(0.8, 1.2)$;
$X \leftarrow$ Redimensionnement(X, ϕ) ;
Sortie : X

(si le facteur de redimensionnement, noté ϕ, est inférieur à 1), soit par extraction d'une sous-image centrale (si ϕ est supérieur à 1). La couleur employée pour le remplissage admet, pour chaque composante (rouge, verte et bleue), la valeur minimale atteinte de la composante correspondante par la sous-image avant rotation.

L'évaluation des modèles porte sur les images entières (*i.e.* non subdivisées selon un quadrillage régulier) des jeux de test à différentes échelles. Les facteurs de redimensionnement employés constituent l'ensemble

$$\{2^{-N} \mid 0 \leq N \leq N_{\max}\} \tag{5.2}$$

Notons que chaque jeu de test admet une valeur $N_{\max}$ qui lui est propre. De plus, il convient de souligner

que la borne inférieure de l'inégalité est 0, et non $N_{\min}$ comme pour l'apprentissage. Ce choix est motivé par le fait que la connaissance de $N_{\min}$ ne peut pas être facilement obtenue en situation opérationnelle, lorsque l'ouvrage à analyser est dépourvu d'annotations. Ainsi, même si nous disposons de cette valeur pour les jeux de test utilisés, il demeure intéressant d'analyser également le comportement des modèles appris pour l'ensemble des facteurs de redimensionnement plus petits que $N_{\min}$. La situation n'est pas symétrique pour $N_{\max}$, qui ne dépend que de la dimension des images considérées.

Par ailleurs, nous évaluons l'intérêt que représente la fusion des prédictions à ces mêmes facteurs de redimensionnement. Formellement, en notant M le modèle appris et R_α l'application qui redimensionne l'image qui lui est passée en paramètre d'un facteur $\alpha \in \mathbb{R}_+^*$, le modèle M' réalisant cette fusion est défini par

$$M'(x) = \text{softmax}\left(\sum_{i=0}^{N_{\max}} R_{2^i}\left(M\left(R_{2^{-i}}(x)\right)\right)\right) \tag{5.3}$$

Ce modèle sera par la suite désigné par **modèle multi-échelles**. La vérité terrain associée à ce modèle est la vérité terrain non redimensionnée, à l'échelle originale.

5.1.2.2 Résultats

La figure 5.5 présente l'évolution de l'exactitude pondérée et du F_1-score sur le jeu de validation du modèle appris sur CODEBRIM. On peut y voir que ces deux séries atteignent simultanément leur maximum pour l'époque 598 avec une valeur de 85.33% pour le F_1-score et 91.14% pour l'exactitude pondérée. Les deux critères de sélection du modèle ont donc désigné la même instance pour l'évaluation. Les résultats quantitatifs de l'ensemble de ces modèles sont présentés dans le tableau 5.5.

D'une manière générale, on peut voir à travers les résultats quantitatifs qu'à l'exception de Rive-de-Gier, les résultats s'améliorent lorsque la dimension des images décroît. C'est, en particulier, le cas pour le rappel, qui est inférieur à 20% pour le tunnel piéton et CODEBRIM lorsque les images sont considérées dans leur résolution d'origine, et qui augmente lorsque les images sont réduites, s'établissant à plus de 85% pour le facteur de redimensionnement le plus petit de chacun des jeux. Sur le tunnel piéton et CODEBRIM, la précision s'accroit également, mais dans une moindre mesure. En conséquence, le F_1-score calculé pour ces deux sites est maximal lorsque la résolution est minimale. Cela peut s'expliquer par un champ réceptif possiblement trop faible du réseau de neurones, qui rend la détection des anomalies, dont beaucoup sont acquises en plan rapproché. De façon analogue à la problématique d'incohérence des annotations, le modèle ne dispose alors pas de suffisamment de contexte visuel pour certaines zones de l'image lorsque cette dernière est fortement résolue.

Tunnel de Rive-de-Gier Pour le tunnel de Rive-de-Gier, les deux échelles testées donnent des résultats semblables. Concernant les métriques pixelliques,

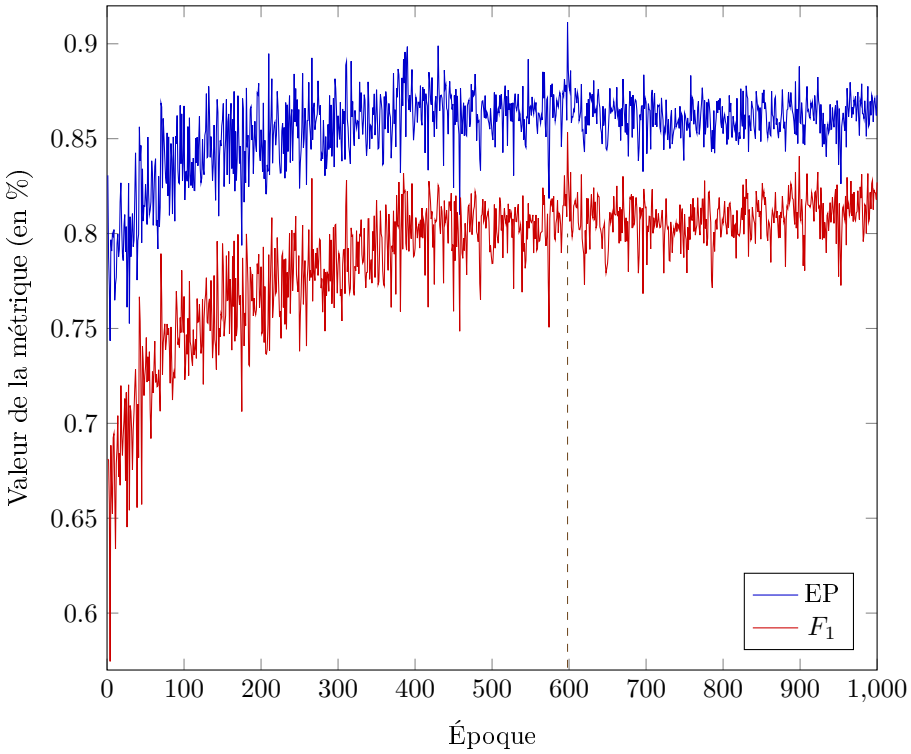


FIGURE 5.5 – Exactitude pondérée et F_1-score des différents modèles générés durant l'apprentissage. La droite verticale en pointillé indique l'époque 598, époque à l'issue de laquelle les deux métriques suivies atteignent leur maximum sur l'apprentissage considéré.

les exactitudes pondérées sont proches pour les deux échelles et avoisinent 90%. On peut toutefois noter une légère tendance à la sur-détection à l'échelle 2^{-1} par rapport à l'échelle originale (la précision est plus faible et le rappel plus élevé pour le premier facteur de redimensionnement que pour le second). Concernant le jeu de test restreint, le retrait des images dépourvues d'anomalies réduit mécaniquement le nombre de fausses alarmes tout en laissant le nombre de vrais positifs et le nombre de faux négatifs inchangés. La précision s'en voit alors tirée vers le haut tandis que le rappel n'évolue pas. Sur ce jeu de test, on observe ainsi une hausse de la précision, et ce, à toutes les échelles. Cette hausse est cependant plus modérée pour le modèle multi-échelles, pour lequel la précision passe de 54.17% à 63.32%, traduisant ainsi le fait que le modèle multi-échelles génère moins de fausses alarmes que les stratégies mono-échelles sur l'ensemble du tunnel.

En termes de composantes connexes (*cf.* figure 5.6), on constate que les fers apparents sont globalement bien détectés. Par exemple, pour un seuil d'admission de 50%, on détecte 65.71% des fers apparents à l'échelle 1 et 71.43%

	EP	P	R	F_1
Tunnel piéton ($\times 1$)	54.80	33.32	10.18	15.60
Tunnel piéton ($\times 2^{-1}$)	77.64	37.79	57.97	45.75
Tunnel piéton ($\times 2^{-2}$)	88.80	37.93	81.34	51.73
Tunnel piéton ($\times 2^{-3}$)	**91.81**	43.25	**86.82**	**57.73**
Tunnel piéton (Multi-échelles)	68.40	**61.69**	37.47	46.62
Rive-de-Gier ($\times 1$)	86.00	44.52	72.08	55.04
Rive-de-Gier ($\times 2^{-1}$)	**91.61**	32.89	**83.38**	47.18
Rive-de-Gier (Multi-échelles)	90.43	**54.17**	80.92	**64.90**
Rive-de-Gier (R) ($\times 1$)	85.79	**68.38**	72.08	70.18
Rive-de-Gier (R) ($\times 2^{-1}$)	**91.18**	54.31	**83.38**	65.77
Rive-de-Gier (R) (Multi-échelles)	90.11	63.32	80.92	**71.05**
CODEBRIM ($\times 1$)	55.61	52.39	16.02	24.53
CODEBRIM ($\times 2^{-1}$)	78.11	72.56	64.23	68.15
CODEBRIM ($\times 2^{-2}$)	**88.79**	75.92	**86.63**	**80.92**
CODEBRIM (Multi-échelles)	79.86	**84.68**	63.52	72.59

TABLEAU 5.5 – Scores obtenus (en %) sur le jeu de test pour la segmentation de fers apparents (EP : exactitude pondérée ; P : précision ; R : rappel ; F_1 : F_1-score). Chaque modèle est évalué sur le site ayant servi à son apprentissage. Le jeu Rive-de-Gier (R) correspond au jeu de test restreint de Rive-de-Gier. Le coefficient multiplicateur indiqué entre parenthèses renseigne le facteur de redimensionnement utilisé sur le jeu de test.

à l'échelle 2^{-1}. Trois exemples de prédictions sont montrés en figure 5.7. On remarque que la tendance précédemment conjecturée se confirme : le modèle tend à sous-détecter à l'échelle originale tandis qu'il génère des composantes légèrement trop grandes, comprenant alors des fausses alarmes, à l'échelle 2^{-1}. En tant que tel, ce phénomène semble être le seul fruit du hasard et nous n'avons trouvé aucun élément pouvant expliquer sa cause. Toutefois, on peut émettre l'hypothèse que la valeur attribuée à $N_{\min}$ est sur-évaluée et qu'un apprentissage réalisé à l'échelle originale n'aurait pas représenté de risque quant à la cohérence des vérités terrain telles qu'employées lors de l'apprentissage (*cf.* discussion en section « méthodologie »).

Le réseau semble donc avoir un comportement proche selon les deux échelles testées et le modèle multi-échelles n'a, en conséquence, pas une grande incidence sur l'aspect qualitatif des prédictions.

Tunnel piéton Pour le tunnel piéton, on constate en premier lieu que la prédiction à l'échelle originale donne de mauvais résultats. En effet, on y relève une exactitude pondérée inférieure à 55% et un F_1-score de 16%. La principale cause de cette faible performance est à chercher du côté du rappel qui est, lui aussi, très bas, avec seulement 10% des pixels représentant des fers apparents qui sont bien classés par le modèle. On constate alors que les résultats s'améliorent au fur et à mesure que la résolution décroît, pour atteindre plus de 90%

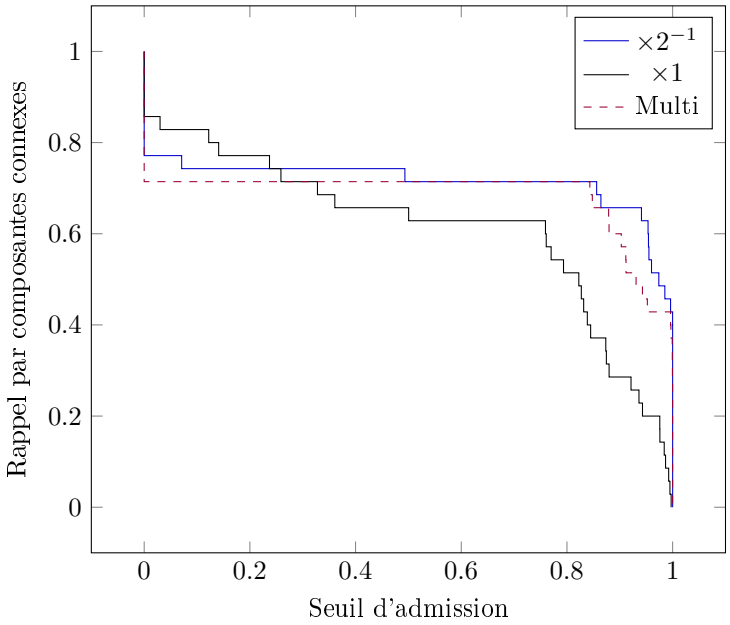
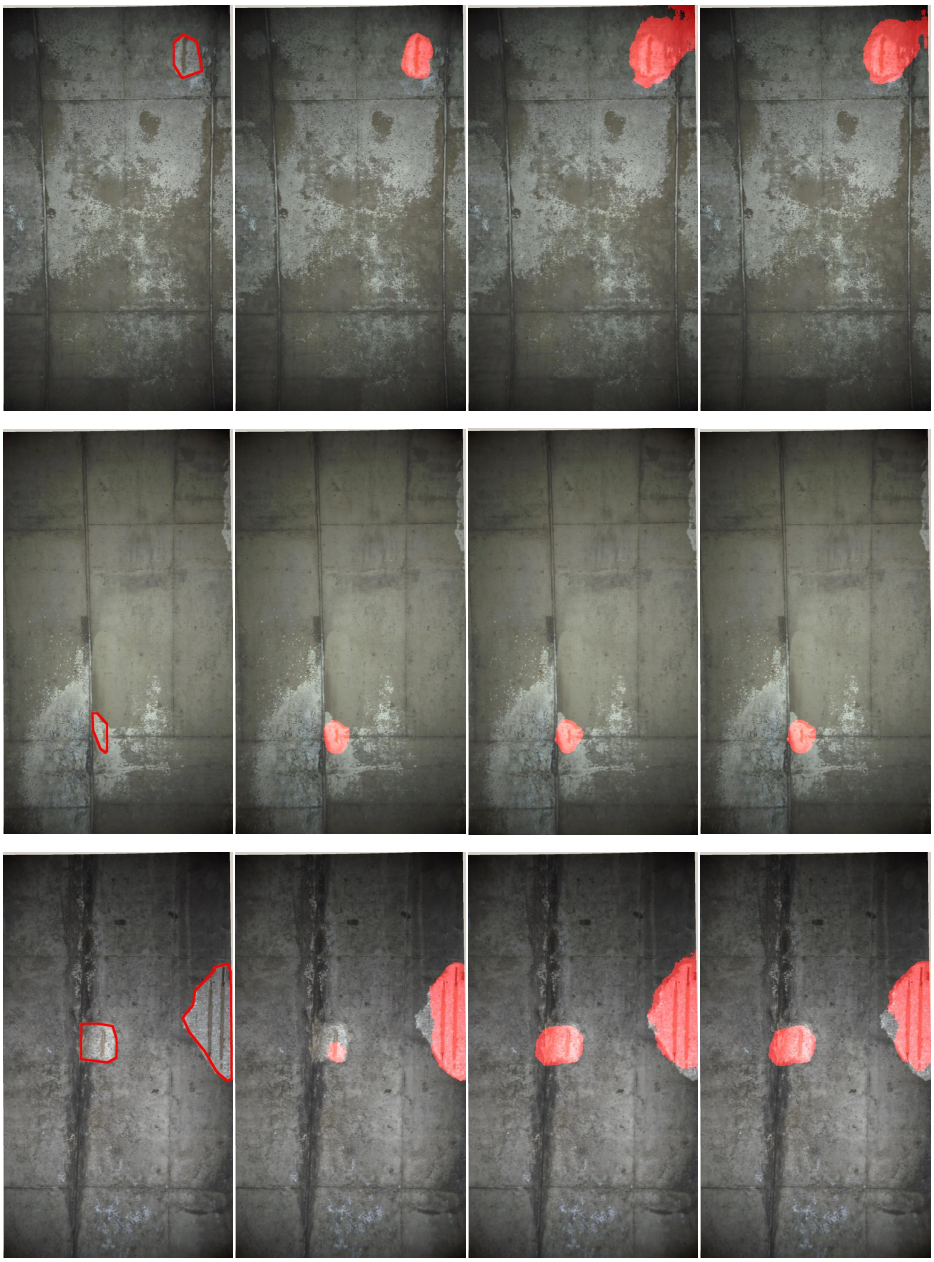


FIGURE 5.6 – Courbes de rappel par composantes connexes obtenues sur le tunnel de Rive-de-Gier

d'exactitude pondérée et près de 60% de F_1-score pour la résolution la plus basse.

L'analyse des courbes de rappel par composantes connexes (*cf.* figure 5.8) nous permet de déduire que, pour près de 40% des fers apparents, aucun pixel les constituant n'est prédit comme anomalie par le modèle à l'échelle 1. À l'échelle 2^{-1}, on détecte environ 50% des composantes connexes lorsque le seuil d'admission est fixé à 50%. Pour les échelles 2^{-2} et 2^{-3}, la différence de rappel n'est notable que pour des seuils d'admission inférieurs à 60%, pour lesquels le rappel par composantes connexes à l'échelle 2^{-2} est supérieur que ce même rappel à l'échelle 2^{-3}. Ainsi, à l'échelle 2^{-2}, on détecte un peu plus de composantes connexes qu'à l'échelle 2^{-3}, mais il s'agit alors de détections partielles. Les exemples de prédictions donnés en figure 5.9 nous permettent de constater que des fausses alarmes sont présentes à tous les facteurs d'échelle. Ainsi, certains végétaux ou graffitis se retrouvent, à tort, considérés comme fers apparents. C'est aussi le cas pour une partie du sol ou du revêtement dans les zones sombres et granuleuses. En cohérence avec les chiffres du rappel par composantes connexes, les fers apparents sont de mieux en mieux détectés aux deux plus faibles facteurs d'échelle, 2^{-2} et 2^{-3}. Concernant la différence entre ces deux échelles, une composante connexe de plus est reconnue dans la seconde mais les prédictions y sont alors spatialement moins bien délimitées que dans la première.

Sur le tunnel piéton, on mesure que le modèle multi-échelles a un effet



Vérité terrain ×1 $\times 2^{-1}$ Multi-échelles

FIGURE 5.7 – Résultats obtenus pour le tunnel de Rive-de-Gier selon deux facteurs d'échelle (les fers apparents sont détourés en rouge dans la vérité terrain).

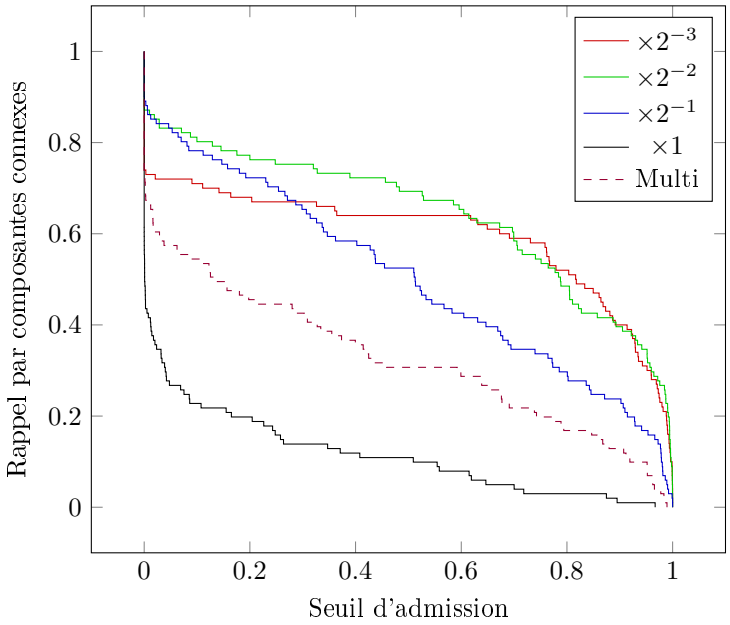


FIGURE 5.8 – Courbes de rappel par composantes connexes obtenues sur le tunnel piéton

positif sur la qualité des prédictions. En effet, par rapport au modèle prédisant uniquement sur les images en pleine résolution, ce modèle parvient à multiplier la précision par deux et le rappel par quatre. Pour les autres métriques, les performances du modèle multi-échelles ne sont pas meilleures que celles des modèles mono-échelles. L'ensemble de ces éléments montre que le modèle multi-échelles a tendance à sous-détecter les anomalies en comparaison des modèles mono-échelles.

CODEBRIM Pour CODEBRIM, on observe, comme pour le tunnel piéton, de meilleurs résultats, toutes métriques confondues, pour les résolutions les plus basses. Les métriques qui bénéficient le plus de cette hausse sont l'exactitude pondérée et le rappel. Ainsi, on en déduit que le modèle a tendance à sous-détecter à l'échelle originale.

Au niveau du rappel par composantes connexes (*cf.* figure 5.10), on constate qu'il augmente significativement lorsque le facteur de redimensionnement diminue. À l'échelle originale, on reconnait 40% des composantes connexes lorsque le seuil d'admission est de 50% et environ 15% des fers apparents ne sont jamais détectés à cette échelle. Pour ce même seuil d'admission, on parvient à repérer respectivement 80% et 90% aux échelles 2^{-1} et 2^{-2}. Deux exemples de prédictions sont présentés en figure 5.11. Sur l'image de gauche, on constate que le modèle semble perturbé par la démarcation entre la zone éclairée et la partie ombragée de l'image aux résolutions les plus hautes. De plus, quelques fausses

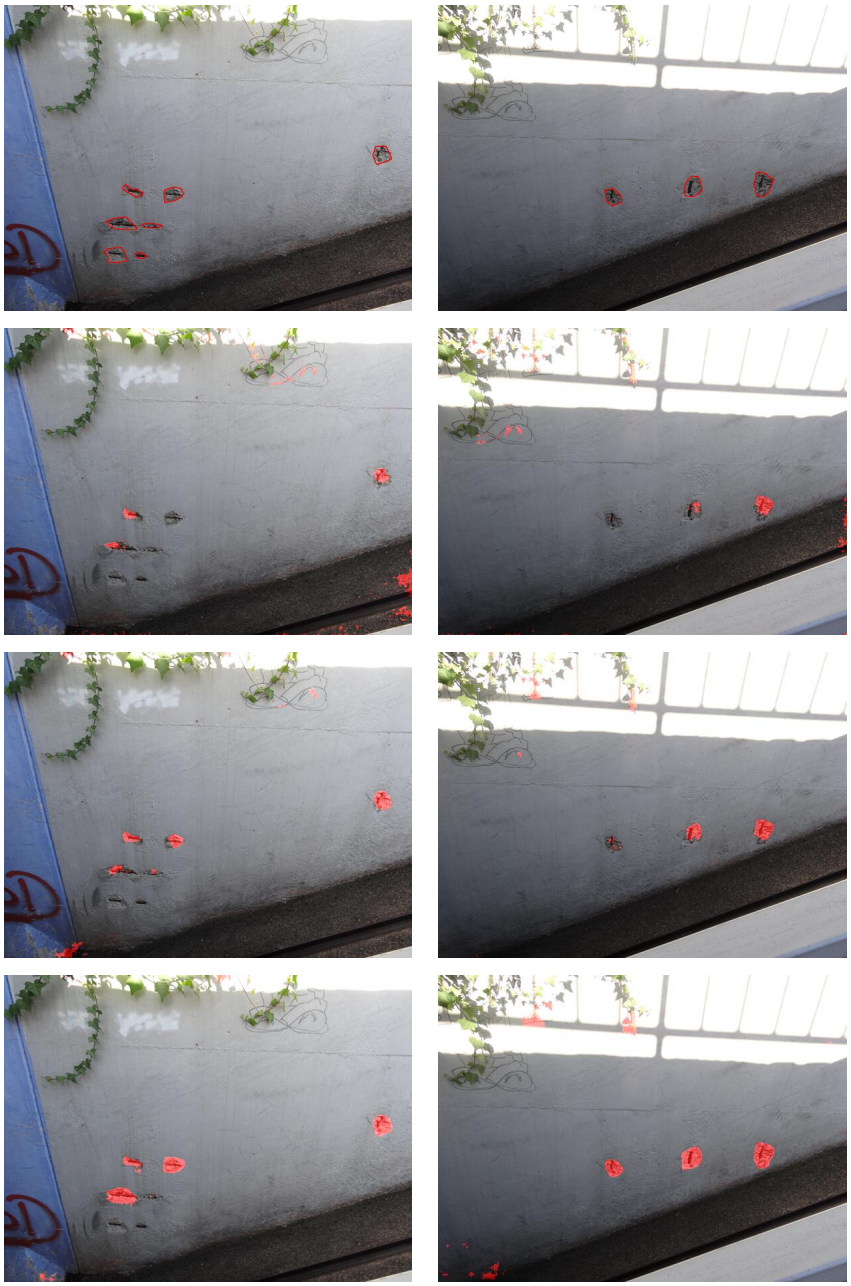


FIGURE 5.9 – Résultats obtenus pour le tunnel piéton selon quatre facteurs d'échelle – **1ère ligne :** vérité terrain (les fers apparents sont détourés en rouge) ; **2ème ligne :** prédiction à l'échelle $\times 1$; **3ème ligne :** prédiction à l'échelle $\times 2^{-1}$; **4ème ligne :** prédiction à l'échelle $\times 2^{-2}$.

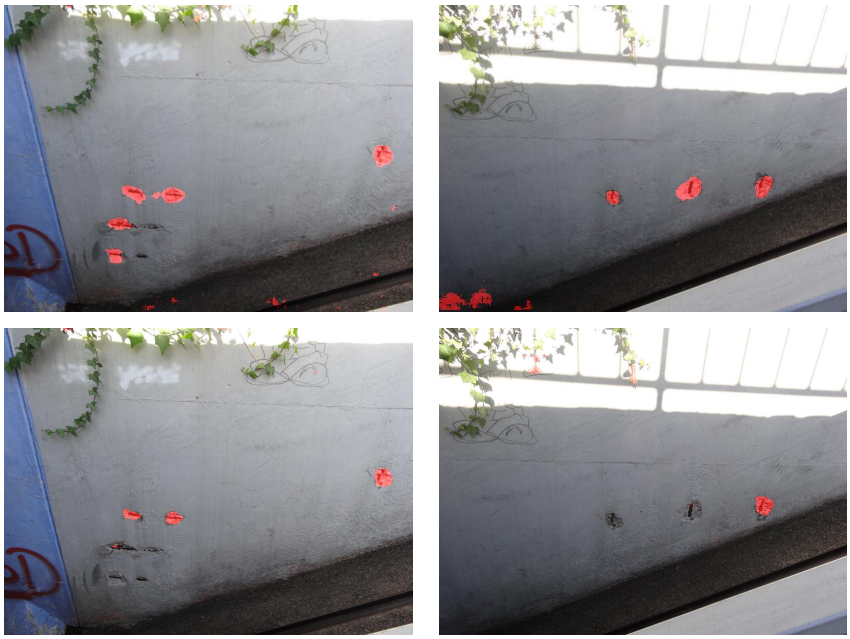


FIGURE 5.9 – Résultats obtenus pour le tunnel piéton selon quatre facteurs d'échelle – **1ère ligne :** prédiction à l'échelle $\times 2^{-3}$; **2ème ligne :** prédiction multi-échelles.

alarmes se situent sur le revêtement granuleux en haut de l'image. Ces imperfections s'estompent lorsque la résolution baisse, jusqu'à produire un résultat presque parfait à l'échelle 2^{-2}. Dans ce dernier cas, le relevé proposé par le modèle est spatialement mieux délimité que le polygone englobant définissant la vérité terrain pour cet exemple. Sur l'image de droite, on peut voir que deux fers apparents ne sont pas détectés à l'échelle originale, alors qu'ils le sont aux échelles 2^{-1} et 2^{-2}, au prix d'un plus grand nombre de faux positifs. Ces faux positifs se situent essentiellement sur une échelle métallique qui est posée sur un des piliers du pont.

Qualitativement, le modèle multi-échelles génère de bien meilleures prédictions que son homologue opérant sur les images en pleine résolution. En effet, sur l'image de gauche, on peut voir que l'anomalie est presque parfaitement délimitée et qu'aucune fausse alarme n'est présente. Sur l'exemple de droite, tous les fers apparents sont correctement identifiés et l'échelle métallique posée sur l'ouvrage ne compte plus que partiellement parmi les fausses alarmes, comme pour la prédiction du modèle à l'échelle 2^{-1}.

Synthèse Dans cette expérimentation, nous avons étudié la capacité des réseaux de neurones à modéliser la variabilité d'aspects des anomalies et des revêtements sains au sein d'un même ouvrage. Trois sites ont été considérées,

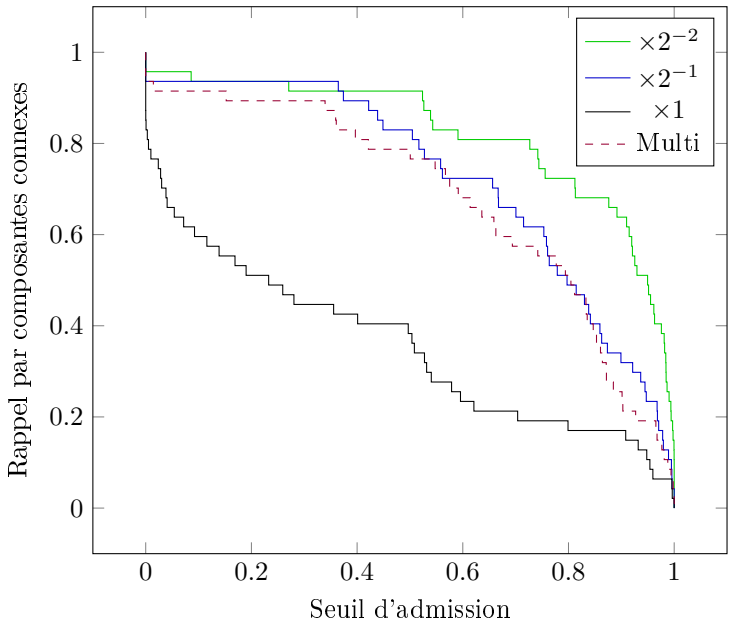


FIGURE 5.10 – Courbes de rappel par composantes connexes obtenues sur la base CODEBRIM

FIGURE 5.11 – Résultats obtenus pour la base CODEBRIM selon trois facteurs d'échelle – **1ère ligne :** vérité terrain (les fers apparents sont détourés en rouge).

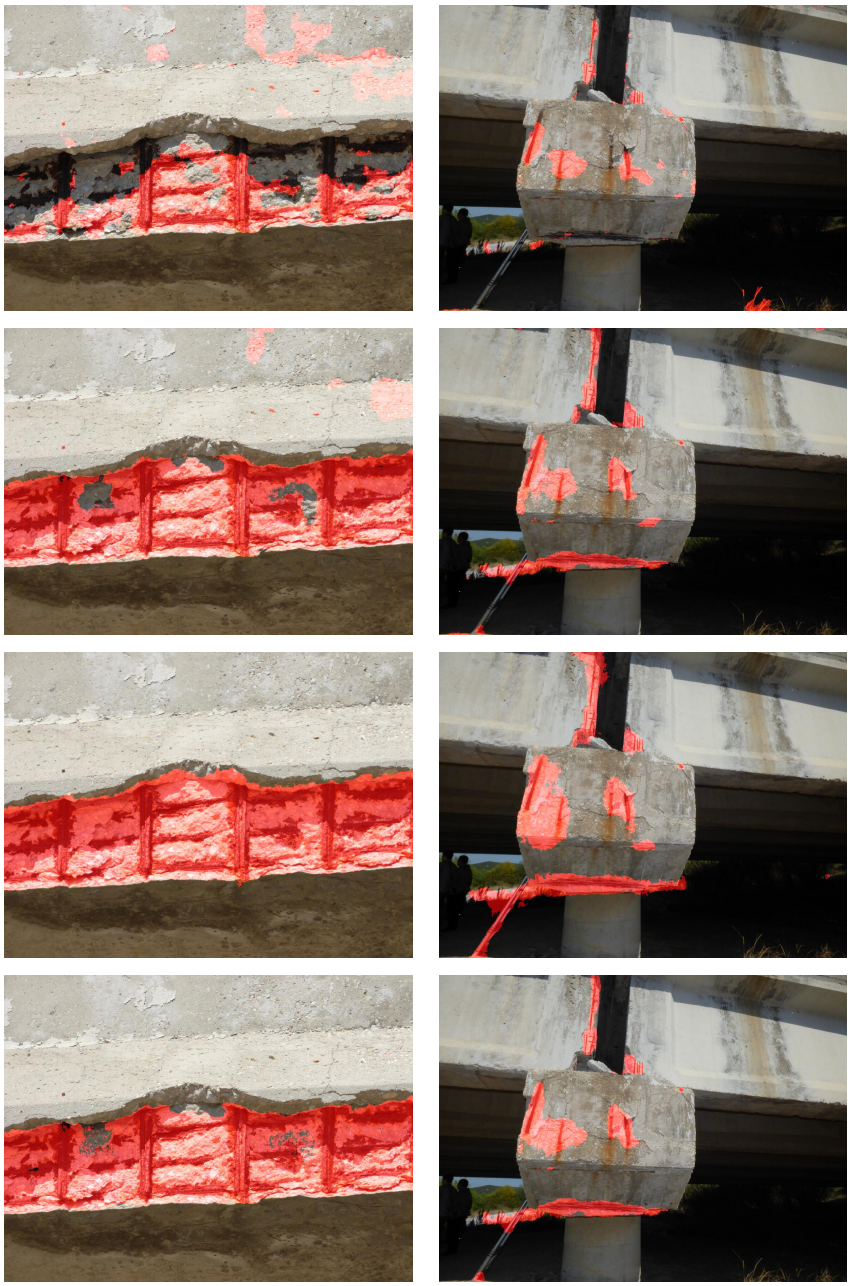


FIGURE 5.11 – Résultats obtenus pour la base CODEBRIM selon trois facteurs d'échelle – **1ère ligne :** prédiction à l'échelle $\times 1$; **2ème ligne :** prédiction à l'échelle $\times 2^{-1}$; **3ème ligne :** prédiction à l'échelle $\times 2^{-2}$; **4ème ligne :** prédiction multi-échelles.

donnant chacun lieu à un apprentissage et à une évaluation. Ces trois expériences ont été réalisés indépendamment, avec des jeux de données adaptés à chaque site. Cette expérimentation nous a permis de mettre au point des modèles capables de détecter 80% des composantes connexes constituant les anomalies (avec un seuil d'admission de 50%) tout en ayant une précision pixellique supérieure à 40%. Cependant, ces résultats sont obtenus sur des images redimensionnées, les prédictions à l'échelle originale demeurent, pour le tunnel piéton et CODEBRIM, significativement en deçà, possiblement à cause d'un champ réceptif trop limité du modèle. L'utilisation de modèles multi-échelles constitue une première réponse à cette problématique puisque ces modèles permettent des cartographies de meilleure qualité (toutes métriques confondues) que les modèles mono-échelles à l'échelle originale. Une autre limitation des modèles mis en œuvre est qu'ils demandent un apprentissage sur une portion significative de l'ouvrage à inspecter (au moins 60% des données de l'ouvrage). Ainsi, il est nécessaire d'annoter une grande quantité de données pour convenablement paramétrer ces modèles, ce qui est souvent irréaliste en pratique. Nous étudions les capacités de généralisation des modèles entre différents sites au chapitre 6 du présent manuscrit.

Bien que ce travail n'ait pas donné lieu à une valorisation scientifique, certains travaux préalables, portant notamment sur l'influence du facteur de redimensionnement sur les résultats d'un modèle U-Net [5], ont été publiés en conférence internationale [145].

5.2 Relevés LCMS

Pour les données LCMS, nous explorons des stratégies de cartographie de fissures par quadrillage régulier et par segmentation sémantique, à l'instar du travail réalisé pour la reconnaissance des anomalies sur les images photographiques. Deux spécificités sont cependant à relever pour les expérimentations sur les données LCMS. Tout d'abord, nous ne travaillons qu'avec les images en pleine résolution, sans recourir aux stratégies multi-échelles développées à la section précédente. Cet élément est motivé par la nature des anomalies considérées, qui sont des fissures relativement fines. Un sous-échantillonnage risquerait de faire partiellement disparaître certaines fissures et de rompre la connexité des vérités terrain associées à ces dernières. Ensuite, les données LCMS étant multimodales, nous réalisons une évaluation comparée de différents traitements de ces modalités ainsi que de l'apport de leur utilisation conjointe au sein d'un réseau de neurones.

Dans le chapitre 2, il a été vu que la normalisation de la profondeur des données LCMS était un sujet non trivial dans le sens où les données brutes manquent fortement de dynamique et qu'il n'existe pas de traitement « naturel » permettant de la corriger. Différentes normalisations ont été proposées et un des objectifs que l'on cherche à atteindre est de déterminer si l'une d'elles se prête mieux à la cartographie des anomalies qu'une autre. À cette fin, nous réalisons un comparatif de plusieurs configurations de données d'entrées. Chaque expérimentation mise en œuvre est donc répliquée pour chacune de ces configurations.

Pour clarifier les notations, on introduit une nomenclature pour désigner les configurations. L'utilisation conjointe de deux canaux A et B est notée $A+B$. Les canaux utilisés ont été décrits dans le chapitre 2, section 2.2.3, et sont les suivants :

- I : Intensité
- D_r : Profondeur brute
- D_c : Profondeur centrée
- D_a : Profondeur ajustée par régression robuste
- B : Carte des *outliers*.
- L : Carte booléenne indiquant l'emplacement des points hors de portée de la composante de profondeur (0 pour les points hors de portée et 1 pour les autres points).

Par exemple, « $I + D_r$ » désigne la configuration où sont simultanément utilisées l'intensité et la profondeur brute. S'agissant de la fusion des différentes modalités, nous procédons systématiquement par *early fusion* (voir chapitre 3, section 3.1.5), c'est-à-dire que les composantes des données LCMS sont concaténées selon l'axe des canaux avant d'être présentées au réseau. Les configurations à l'étude sont reportées dans le tableau 5.6.

Pour l'augmentation de données, nous reprenons la méthode de Drouyer [130] que nous avons mise en œuvre pour les images photographiques, à ceci

près que nous retirons les étapes de rotation et de changement d'échelle afin de préserver la cohérence de nos vérités terrain. En effet, comme les données LCMS sont consacrées à la reconnaissance de fissures et que notre processus de labellisation pour ce type d'anomalie produit des annotations ne faisant qu'un seul pixel de large (*cf.* chapitre 2, section 2.2.4), l'application de ces deux traitements peut morceler ces annotations, représentant principalement de longues fissures, en plusieurs composantes connexes de petite taille, l'interpolation étant faite au plus proche voisin pour éviter la création d'artefacts. Le risque, en gardant ces étapes, est alors de voir le réseau considérer, à tort, les petites imperfections ou pertes de matières comme fissures dès lors qu'elles ont une apparence filiforme. Il a donc été décidé de les supprimer en ne conservant que la partie relative à la normalisation et les miroirs horizontaux et verticaux. La méthode modifiée est décrite dans l'algorithme 2.

I	
$I + D_r$	$I + D_r + L$
D_r	$D_r + L$
$I + D_c$	$I + D_c + L$
D_c	$D_c + L$
$I + D_a$	$I + D_a + B$
D_a	$D_a + B$

TABLEAU 5.6 – Ensemble des treize configurations testées pour la cartographie des anomalies

Algorithme 2 : Algorithme d'augmentation de données de Drouyer [130] après retrait des opérations de rotation et de redimensionnement (voir texte).
$\mathcal{U}$ désigne la loi uniforme continue.

Entrée : La sous-image à traiter X
$\alpha \sim \mathcal{U}(0.5, 1.5)$;
$\beta \sim \mathcal{U}(-0.15, 0.15)$;
$X \leftarrow \alpha(X - \bar{X}) + \bar{X} + \beta$ (où $\bar{X}$ est la moyenne des valeurs de X) ;

$X \leftarrow$ Miroir_horizontal(X) avec une probabilité de $\frac{1}{2}$;
$X \leftarrow$ Miroir_vertical(X) avec une probabilité de $\frac{1}{2}$;
Sortie : X

5.2.1 Cartographie par quadrillage régulier

Résumé du protocole expérimental

- **Architecture** ResNet-18
- **Poids initiaux** Aléatoires
- **Époques** 1000
- **Optimiseur** *Adam*
- **Augmentation de données** Drouyer modifié (algorithme 2)
- **Critère d'optimisation** Entropie croisée pondérée
- **Critères de sélection des modèles à évaluer**
  - Modèle maximisant le F_1-score sur le jeu de validation
  - Modèle maximisant l'exactitude pondérée sur ce même jeu
- **Objectifs**
  - Évaluer l'influence des différentes modalités des données LCMS sur les cartographies prédites
  - Comparer deux stratégies de cartographie par quadrillage régulier
  - Comparer deux critères de sélection du modèle dans un contexte de fort déséquilibre entre classes

5.2.1.1 Méthodologie

Pour la cartographie par quadrillage régulier sur données LCMS, nous évaluons et comparons deux approches.

Stratégie mono-grille et cartographie sans recouvrement : Il s'agit de la méthode de cartographie par quadrillage régulier « classique » que nous avons décrite dans le chapitre 3 et qui a été mise en œuvre dans la section 5.1.1.

Stratégie multi-grilles et cartographie avec recouvrement : Nous implémentons et évaluons la méthode décrite par Cha *et al.* [9] et présentée dans l'état de l'art (*cf.* chapitre 4, section 4.2.1).

Pour ces deux méthodes, des sous-images de 256×256 pixels sont extraites selon un quadrillage régulier. Pour la stratégie mono-grille, c'est un quadrillage aligné sur les bords haut et gauche qui est utilisé. La stratégie multi-grilles fait de même mais considère également un quadrillage de mêmes dimensions avec un décalage d'une demi-imagette selon chacun des axes. De plus, les sous-images qui présentent une fissure excentrée, plus difficile à détecter du fait de l'insuffisance de contexte visuel, ne sont alors pas conservées dans les jeux (apprentissage, validation et test) par cette seconde approche. La figure 5.12 présente ces deux quadrillages sur un exemple et la figure 5.13 résume l'approche de sélection des sous-images de cette méthode.

La composition des jeux de données est décrite dans le tableau 5.7. On constate que l'utilisation de deux quadrillages pour l'extraction des sous-images permet d'obtenir des jeux de données plus grands (environ deux fois plus

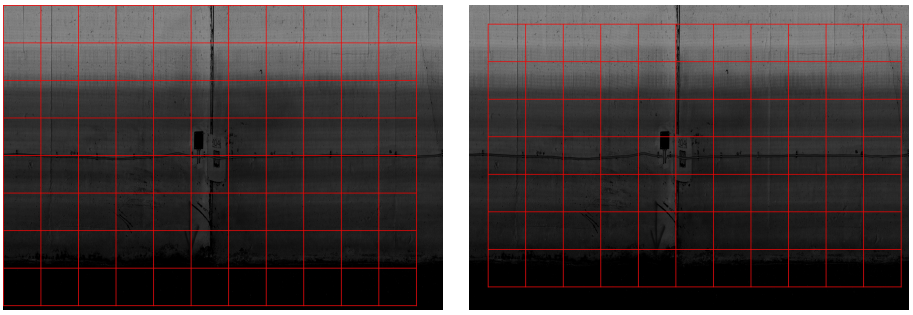
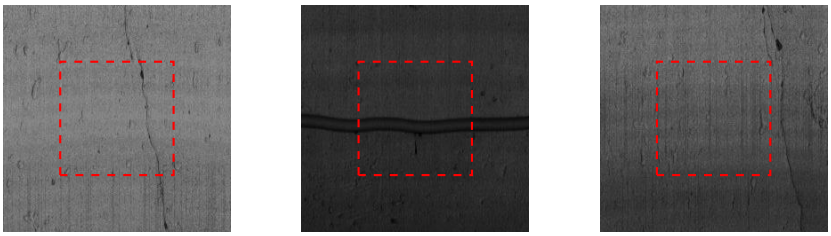


FIGURE 5.12 – Quadrillages utilisés pour extraire les sous-images (de résolution 256×256). La stratégie mono-grille emploie celui de gauche tandis que la stratégie multi-grilles considère simultanément les deux quadrillages.



(a) (b) (c)

FIGURE 5.13 – Illustration de l'approche de sélection des sous-images pour la stratégie multi-grilles. Toute sous-image comprenant une ou plusieurs fissures et dont aucune n'intersecte la zone centrale de la sous-image, matérialisée ici par un carré rouge, est rejetée. Ainsi, les sous-images (a) et (b) sont conservées, puisque la première présente une fissure centrée et que la seconde est un revêtement sain mais la sous-image (c) est rejetée et n'est donc pas retenue pour la constitution des jeux.

d'exemples) et, ce, malgré le rejet des exemples présentant des fissures excentrées. Toutefois, ce procédé d'éviction réduit la proportion de sous-images de classe « fissure » presque de moitié pour l'ensemble des jeux. La quantité d'exemples de fissures diminue légèrement, mais chacune des fissures est alors centrée sur sa sous-image, simplifiant la tâche à apprendre.

	Apprentissage	Validation	Test
Stratégie mono-grille	2640 *(5.57%)*	2640 *(7.80%)*	14960 *(6.30%)*
Stratégie multi-grilles	4805 *(3.00%)*	4729 *(3.93%)*	27088 *(3.23%)*

TABLEAU 5.7 – Composition des jeux de données LCMS utilisés pour la cartographie des fissures par classification (en nombre de sous-images). Le pourcentage de sous-images présentant une fissure est indiqué entre parenthèses.

	Critère de sélection EP				Critère de sélection F_1			
	EP	P	R	F_1	EP	P	R	F_1
D_a	51.00	6.53	52.12	11.61	51.0	6.53	**52.12**	11.61
$D_a + B$	74.35	29.84	57.85	39.37	69.83	60.24	41.50	49.15
D_c	59.35	14.89	30.36	19.98	58.45	26.45	20.80	23.29
$D_c + L$	61.61	27.83	28.13	27.98	61.61	27.83	28.13	27.98
D_r	63.32	17.79	38.64	24.37	61.23	26.89	27.49	27.19
$D_r + L$	62.26	18.98	34.39	24.46	59.96	26.36	24.52	25.41
I	74.02	27.68	58.28	37.53	69.29	58.14	40.55	47.77
$I + D_a$	74.81	35.36	56.58	43.52	69.00	56.08	40.12	46.78
$I + D_a + B$	72.47	**41.88**	49.57	45.40	69.80	55.96	41.82	47.87
$I + D_c$	76.23	22.87	**67.83**	34.20	68.30	45.56	39.80	42.49
$I + D_c + L$	70.51	23.99	52.12	32.86	70.15	51.85	42.99	47.01
$I + D_r$	73.98	30.19	56.79	39.42	70.18	**61.55**	42.14	50.03
$I + D_r + L$	**77.83**	41.65	61.46	**49.65**	**73.73**	58.00	49.89	**53.68**

(a) Stratégie mono-grille

	Critère de sélection EP				Critère de sélection F_1			
	EP	P	R	F_1	EP	P	R	F_1
D_a	81.79	26.48	70.14	38.45	77.07	**83.63**	54.5	**65.99**
$D_a + B$	82.79	25.21	72.86	37.46	74.63	77.06	49.76	60.47
D_c	70.37	20.67	46.74	28.67	65.80	43.78	33.02	37.65
$D_c + L$	65.35	21.12	35.08	26.37	63.52	29.74	29.37	29.55
D_r	66.48	20.66	37.82	26.72	63.60	30.24	29.48	29.86
$D_r + L$	63.81	15.03	34.05	20.86	60.18	27.93	22.28	24.79
I	**86.26**	28.09	**79.31**	41.49	**80.39**	58.16	**62.28**	60.15
$I + D_a$	79.20	**53.3**	60.18	**56.53**	71.97	67.44	44.66	53.74
$I + D_a + B$	76.87	28.67	58.64	38.52	71.34	58.85	43.72	50.16
$I + D_c$	75.20	19.62	58.40	29.37	72.90	42.16	48.00	44.89
$I + D_c + L$	81.73	35.60	67.54	46.62	78.76	69.05	58.40	63.28
$I + D_r$	84.49	31.06	74.51	43.84	77.22	70.82	55.20	62.04
$I + D_r + L$	84.96	32.23	75.20	45.13	78.04	69.84	56.91	62.72

(b) Stratégie multi-grilles

TABLEAU 5.8 – Scores obtenus (en %) sur le jeu de test par les différentes configurations et selon deux stratégies (mono- et multi-grilles) pour la classification de sous-images LCMS (EP : exactitude pondérée ; P : précision ; R : rappel ; F_1 : F_1-score).

Pour chaque configuration testée, et pour chacune des deux approches, la même méthodologie est appliquée : nous entraînons un modèle ResNet-18 [4] (*cf.* chapitre 3, section 3.1.2), sans utiliser de modèles pré-appris (à notre connaissance, il n'existe pas de tels modèles adaptés aux données LCMS) et à l'aide de l'optimiseur Adam pendant 1000 époques. Comme nous disposons d'un jeu de validation, nous évaluons le modèle résultant de chaque époque sur ce dernier. Les modèles ayant réalisé la meilleure exactitude pondérée et le meilleur F_1-score sont alors évalués sur le jeu de test.

5.2.1.2 Résultats

Les résultats quantitatifs sont reportés dans le tableau 5.8.

Généralités De façon générale, on peut relever que les modèles sélectionnés par le critère EP admettent un rappel supérieur à ceux dont la sélection est fondée sur le F_1. Inversement, les modèles sélectionnés par ce dernier critère présentent une précision plus élevée que ceux ayant maximisé l'exactitude pondérée sur le jeu de validation. Par ailleurs, on peut relever que la stratégie multi-grilles permet d'atteindre des scores généralement plus élevés que la stratégie mono-grille. C'est, de façon notable, le cas pour l'exactitude pondérée, qui est systématiquement supérieure pour les modèles appris avec la première stratégie.

Influence des modalités Il ressort de ces résultats que les configurations utilisant la profondeur brute ou centrée sans la composante d'intensité se positionnent en queue de classement, tous critères confondus, et ce pour les deux stratégies mises en œuvre (exception faite de la précision pour la stratégie mono-grille et le critère EP, même si cette précision demeure basse). La profondeur ajustée s'en tire à meilleur compte puisque le rappel des configurations D_a et $D_a + B$ est systématiquement supérieur d'au moins 10% par rapport aux configurations D_c (profondeur centrée), $D_c + L$, D_r (profondeur brute) et $D_r + L$, pour les deux approches et les deux critères considérés. Cette bonne performance se manifeste aussi à travers l'exactitude pondérée, avec un écart cependant plus faible.

Un autre point saillant de ces résultats est que l'influence de la carte des points hors de portée L semble limitée. Même si cette modalité a permis à la configuration $I + D_r + L$ de réaliser la meilleure exactitude pondérée et le meilleur F_1-score pour la stratégie mono-grille et ce, pour les deux critères de sélection, elle ne parvient pas à améliorer significativement les performances (toutes métriques confondues) pour l'apprentissage par la stratégie multi-grilles. Dans une moindre mesure, on constate également une hausse des différentes métriques pour la stratégie mono-grille entre les configurations D_c et $D_c + L$, au profit de cette dernière. On ne constate cependant pas une telle hausse pour la stratégie multi-grilles. Cela pourrait s'expliquer par le fait que les fissures sont parfois bordées de points hors de portée, là où les aspérités du béton, qui admettent une apparence proche des fissures dans la carte d'intensité, en sont systématiquement exemptes. Or, les jeux utilisés pour la stratégie mono-grille conservent les fissures excentrées, dont on a vu que l'absence du contexte spatial les entourant pouvait les rendre difficiles à identifier. La carte des points hors de portée L peut alors aider le modèle à déterminer la nature de ces exemples ambigus. Plus généralement, il est probable que la « simplification » des jeux opérée par la stratégie multi-grilles ait contribué à réduire l'intérêt que revêt L pour la reconnaissance des fissures.

On peut également relever que l'utilisation conjointe de l'intensité avec la profondeur dans sa modalité brute donne de bons résultats pour l'ensemble

des expérimentations, s'établissant à un niveau proche (voire supérieur dans certains cas) de celui des configurations où les profondeurs centrée et ajustée la remplacent. La question se pose alors de savoir si la profondeur brute contribue réellement à la prédiction ou si le réseau ne lui accorde qu'une faible importance, l'efficacité du modèle associé étant alors expliquée par la seule composante d'intensité. S'il est difficile d'apporter une réponse définitive à cette question, on peut toutefois estimer l'intérêt que porte la première couche du réseau de neurones à la composante de profondeur en décomposant les premiers filtres du modèle, une fois ce dernier appris. Formellement, le résultat F_i du i-ème filtre de la première couche du modèle ResNet-18 (on a donc $1 \leq i \leq 64$) est donné, en tout point $(x, y) \in \mathbb{Z}^2$, par

$$F_i(x,y) = \sum_{j=1}^{2} \sum_{k=1}^{W} \sum_{l=1}^{H} I(k,l,j) K(x-k, y-l, j) \tag{5.4}$$

où I est la donnée d'entrée, de résolution $W \times H$, et K le noyau de convolution du i-ème filtre. En explicitant la première somme, il vient

$$\begin{aligned} F_i(x,y) = &\left(\sum_{k=1}^{W} \sum_{l=1}^{H} I(k,l,1) K(x-k, y-l, 1) \right) \\ &+ \left(\sum_{k=1}^{W} \sum_{l=1}^{H} I(k,l,2) K(x-k, y-l, 2) \right) \end{aligned} \tag{5.5}$$

On pose $K_j(x,y) = K(x,y,j)$ pour tout $(x,y) \in \mathbb{Z}^2$ et tout $j \in \{1;2\}$. K_1 opère donc sur la composante d'intensité, K_2 sur la profondeur brute. La figure 5.14 représente le nuage de points dont les coordonnées sont les normes euclidiennes de K_1 et K_2, c'est-à-dire

$$\left(\sqrt{\sum_{k=1}^{W} \sum_{l=1}^{H} K_1(k,l)^2}, \sqrt{\sum_{k=1}^{W} \sum_{l=1}^{H} K_2(k,l)^2} \right) \tag{5.6}$$

On y observe un nuage étendu dépeignant une corrélation négative entre les normes de K_1 et de K_2. Autrement dit, pour tout $j \in \{1;2\}$, plus la norme de K_j est grande, plus celle de K_{3-j} est proche de 0. Or, pour tout $(x,y) \in \mathbb{Z}^2$,

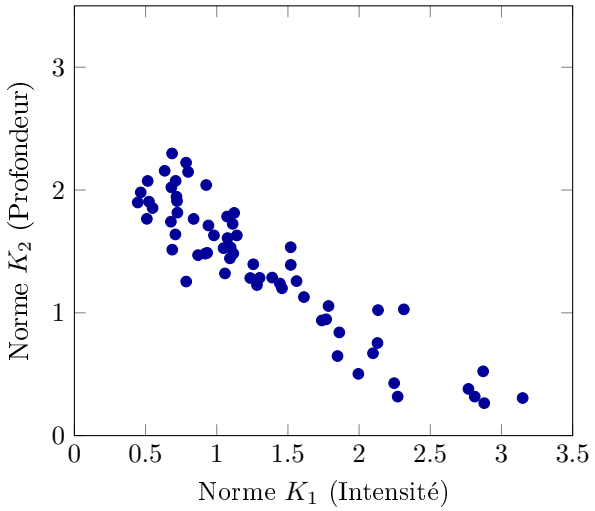


FIGURE 5.14 – Comparaison des normes des noyaux K_1, liés à l'intensité, en fonction des normes des noyaux K_2, relatifs à la composante de profondeur, issus de l'apprentissage réalisé à partir de la configuration $I+D_r$ par la stratégie mono-grille et selon le critère de sélection F_1. Les coordonnées des 64 points sont définies par l'expression 5.6.

on a

$$|(I * K_j)(x,y)| = \left| \sum_{k=1}^{W} \sum_{l=1}^{H} I(k,l,j) K_j(x-k, y-l) \right| \tag{5.7}$$

$$\leq \sum_{k=1}^{W} \sum_{l=1}^{H} |I(k,l,j) K_j(x-k, y-l)| \tag{5.8}$$

$$= \sum_{k=1}^{W} \sum_{l=1}^{H} |I(k,l,j)| \times |K_j(x-k, y-l)| \tag{5.9}$$

$$= \sum_{k=1}^{W} \sum_{l=1}^{H} |I(k,l,j)| \times \sqrt{K_j(x-k, y-l)^2} \tag{5.10}$$

$$\leq \sum_{k=1}^{W} \sum_{l=1}^{H} |I(k,l,j)| \times \|K_j\| \tag{5.11}$$

$$= \|K_j\| \times \left(\sum_{k=1}^{W} \sum_{l=1}^{H} |I(k,l,j)| \right) \tag{5.12}$$

Ainsi, pour tout $(x,y) \in \mathbb{Z}^2$, on a

$$\lim_{\|K_j\| \to 0} (I * K_j)(x,y) = 0 \tag{5.13}$$

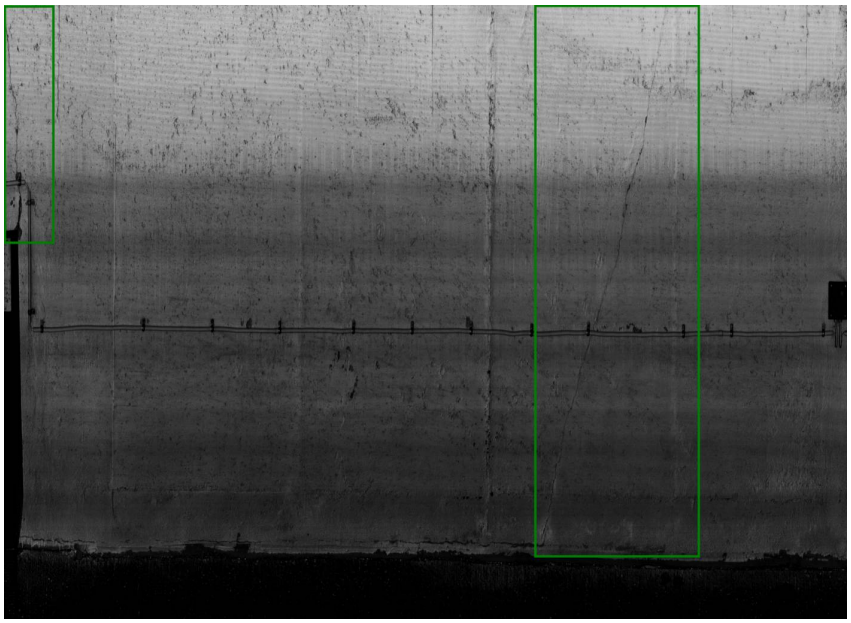
En conséquence, un noyau de faible norme a une faible activité et produit un résultat proche de 0. Les filtres associés sont alors spécialisés pour l'une ou l'autre composante. Le réseau prend donc bien en compte la profondeur brute (au moins au niveau de la première couche de convolution) puisque certains filtres admettent simultanément un noyau opérant sur la composante de profondeur de grande norme tout en ayant un noyau relatif à la composante d'intensité de faible norme.

Par ailleurs, on peut remarquer que la stratégie mono-grille employant la seule profondeur ajustée D_a se traduit par une exactitude pondérée proche de 50% et une précision très basse au regard des résultats obtenus au sein des autres configurations. Cette contre-performance semble être le seul fait du hasard. En effet, les scores atteints par ses configurations voisines ($D_a + B$ pour la stratégie mono-grille, D_a et $D_a + B$ pour la stratégie multi-grilles) sont nettement supérieurs. Également, notons qu'en dehors de ce cas défaillant (configuration D_a, stratégie mono-grille), l'adjonction de B à une configuration ne semble pas permettre d'améliorer significativement les résultats.

Analyse qualitative La figure 5.15 présente des exemples de prédictions pour la configuration I pour les deux approches et les deux critères de sélection du modèle. Pour la stratégie mono-grille, on observe que les deux fissures présentes sont globalement bien détectées pour le critère EP. Néanmoins, on dénombre une dizaine de sous-images classées à tort comme fissure, essentiellement situées au niveau de joints ou sur les rebords de niches, bien que ces derniers éléments soient également présents dans le jeu d'apprentissage. Il n'y a plus qu'une fausse alarme lorsque l'on se base sur le critère F_1, mais la fissure de gauche n'est alors plus détectée. Pour ces deux critères, on note une difficulté à reconnaître les fissures situées en bordure de sous-image. Pour la fissure de droite, une sous-image est ainsi prédite à tort comme saine pour le critère F_1 alors que ce nombre s'élève à deux pour le critère EP. Pour la stratégie multi-grilles, on constate que le critère EP donne également lieu à une dizaine de fausses alarmes. La fissure de gauche est parfaitement reconnue, celle de droite l'est en grande partie. Pour le critère F_1, la fissure de droite est correctement détectée, à l'exception de la sous-image la plus en haut de l'image. En revanche, seul un tiers de la fissure de gauche est effectivement détectée par le réseau de neurones.

Pour les deux stratégies évaluées, on peut noter que les fissures sur le haut de l'image semblent moins bien reconnues. En effet, pour la fissure de droite, la sous-image située en haut de l'image, dans la zone la plus claire de cette dernière, n'est jamais bien détectée parmi ces exemples de prédictions. Ce phénomène pourrait s'expliquer par le fait que seule la partie supérieure des images présente une telle clarté. Les exemples présentant cette même caractéristiques (qu'il s'agisse de fissure ou de revêtement sain) sont donc moins nombreux.

Afin de mieux visualiser les éléments que les réseaux de neurones jugent saillants dans les sous-images, nous avons appliqué la méthode Grad-CAM [19] (*cf.* chapitre 3, section 3.1.2.4), qui permet de localiser les zones de l'image



Carte d'intensité, présentant deux fissures (encadrées en vert)

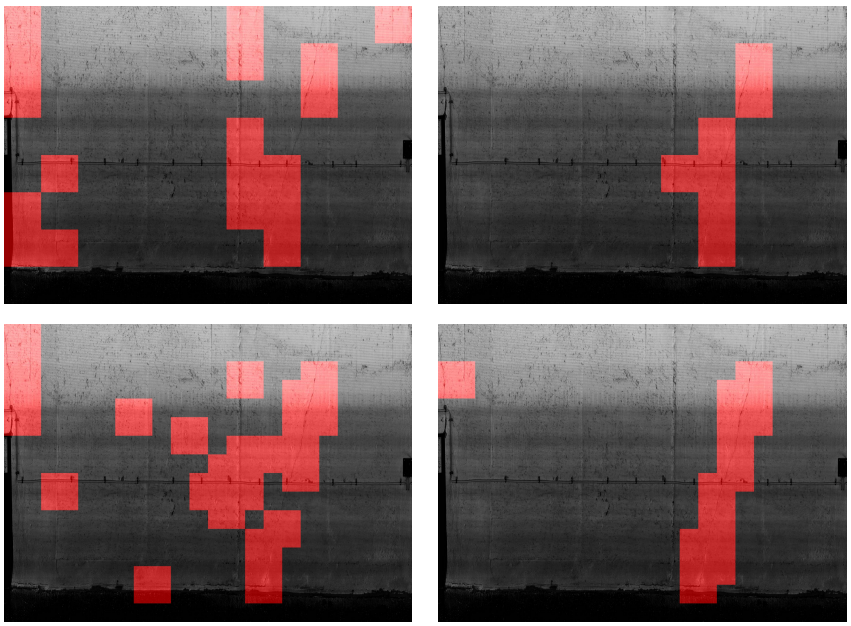
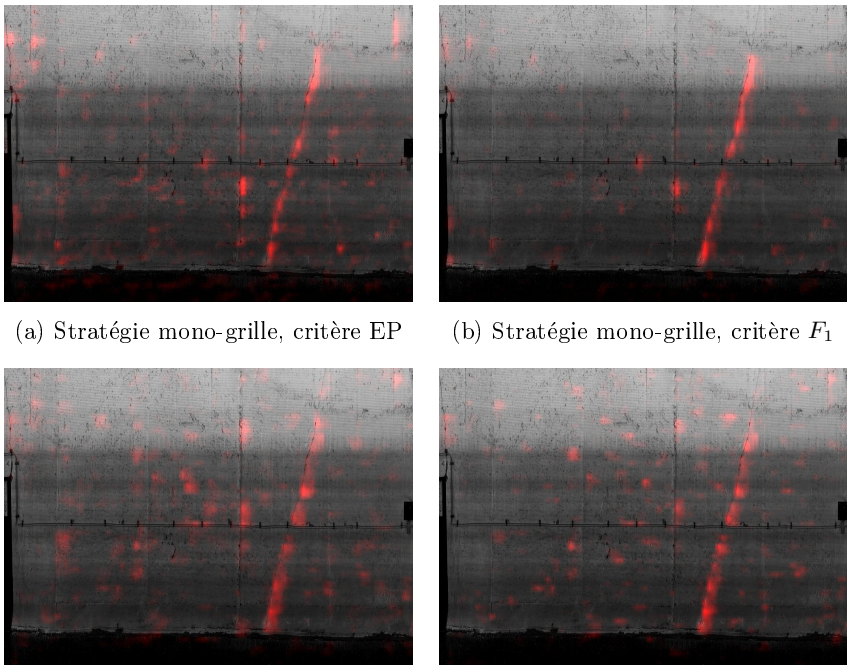


FIGURE 5.15 – Exemples de prédictions réalisées sur une image du jeu de test pour la classification et pour la configuration I. Sur cette image, deux fissures (encadrées en vert) sont à détecter. La première rangée de résultats correspond à la stratégie mono-grille et la seconde présente les prédictions obtenues par la stratégie multi-grilles. La sélection du modèle s'est basée sur l'exactitude pondérée pour la colonne de gauche et sur le F_1-score pour celle de droite.



(a) Stratégie mono-grille, critère EP (b) Stratégie mono-grille, critère F_1

(c) Stratégie multi-grilles, critère EP (d) Stratégie multi-grilles, critère F_1

FIGURE 5.16 – Carte d'activation générée par la méthode Grad-CAM [19] sur la même image de test que la figure 5.15 pour la configuration I.

générant de fortes activations au niveau de la dernière couche de convolution, qui est la plus riche sémantiquement. Les résultats sont reportés en figure 5.16. On y voit une forte zone d'activation au niveau de la fissure de droite et une activation moindre pour celle de gauche, et ce, pour les deux méthodes d'apprentissage, tous critères confondus. De plus, une faible activation au niveau des parties claires de l'image est également notable. À l'exception de la stratégie mono-grille munie du critère F_1, on constate qu'il y a beaucoup d'activations en dehors des voisinages des fissures, souvent au niveau des joints, qui se rapprochent visuellement des fissures.

Synthèse Plusieurs éléments ressortent de cette expérimentation.

Tout d'abord, il apparaît que les modèles sélectionnés par le critère de l'exactitude pondérée (EP) sont, à quelques exceptions près, caractérisés par un nombre très élevés de fausses alarmes, la précision mesurée pour ces modèles plafonnant à environ 53%. À l'inverse, le critère reposant sur le F_1-score permet de réduire drastiquement ce nombre de fausses alarmes, même si cette baisse se fait au prix d'un rappel amoindri.

En ce qui concerne l'analyse différentielle visant à statuer sur l'intérêt des différentes combinaisons de modalités pour la classification des fissures,

nous observons que certaines d'entre elles semblent inefficaces, voire délétères, pour obtenir des modèles présentant de bonnes performances. C'est le cas, par exemples, de la carte des points hors de portée L ou encore de la profondeur centrée D_c. À l'inverse, d'autres modalités, comme l'intensité I ou la profondeur dans sa modalité brute D_r ou ajustée D_a, s'avèrent utiles pour atteindre ce même but.

S'agissant de la comparaison entre les deux approches évaluées, il semble que la stratégie mono-grille conduise à des modèles ayant tendance à être sensiblement moins bons, tant sur le plan qualitatif que quantitatif, que ceux résultant de la stratégie multi-grilles. Bien qu'il convienne de rester prudent, puisque les jeux de test employés pour chacune des approches ne sont pas strictement identiques, on peut néanmoins conjecturer que la stratégie multi-grilles permet l'obtention de jeux facilitant l'apprentissage.

Enfin, nous avons constaté, lors de la conduite de cette expérimentation, que le processus d'optimisation présentait parfois des problèmes de stabilité, la courbe représentant l'évolution du coût mesuré sur le jeu d'apprentissage admettant alors un comportement erratique. Si une seule configuration a été concernée par ce phénomène dans les résultats présentés pour cette expérimentation, il est également survenu à l'occasion de certains des travaux exploratoires qui l'ont précédée, et qui employaient un modèle VGG16 (les résultats de ces travaux n'ont pas été conservés). Il est possible que ce problème soit causé par le nombre insuffisant d'exemples du jeu d'apprentissage au regard de la difficulté de la tâche à apprendre.

5.2.2 Cartographie par segmentation sémantique

Résumé du protocole expérimental

- **Architecture** SegNet
- **Poids initiaux** Aléatoires
- **Époques** 1000
- **Optimiseur** *Adam*
- **Augmentation de données** Drouyer modifié (algorithme 2)
- **Critère d'optimisation** Entropie croisée pondérée
- **Critères de sélection des modèles à évaluer**
  - Modèle maximisant le F_1-score sur le jeu de validation
  - Modèle maximisant l'exactitude pondérée sur ce même jeu
- **Objectifs**
  - Évaluer un modèle de segmentation sémantique pour la détection des fissures
  - Évaluer l'influence des différentes modalités des données LCMS sur les cartographies prédites
  - Comparer deux critères de sélection du modèle dans un contexte de fort déséquilibre entre classes

5.2.2.1 Méthodologie

Nous reprenons l'ensemble des configurations précédentes pour la segmentation sémantique ainsi que les mêmes jeux de données utilisés pour la cartographie par quadrillage régulier sur données LCMS. Nous ne réutilisons cependant pas les jeux construits pour la stratégie multi-grilles. En effet, la présence de recouvrement entre sous-images, apporté par l'utilisation conjointe des deux quadrillages, est moins critique en segmentation sémantique qu'en classification : même excentrée au sein d'une sous-image, une fissure sera généralement représentée par plusieurs dizaines de pixels. Or, l'apprentissage comme l'évaluation sont, ici, réalisés à l'échelle du pixel. Ainsi, les cas réellement problématiques auront un impact négligeable sur le modèle appris, puisque composés de moins de pixels.

Nous ajoutons, de plus, un autre jeu d'évaluation : il s'agit du jeu de test auquel on retire toutes les sous-images ne représentant que des revêtements sains. De cette façon, nous nous rapprochons des conditions d'évaluation de nombreux travaux sur la segmentation sémantique des fissures pour lesquels tous les exemples annotés comportent des fissures. Nous parlons de jeu « tout venant » pour le jeu de test non modifié et de jeu « restreint » pour celui dont toutes les sous-images présentent une fissure. Notons que ce jeu de test restreint correspond exactement à l'ensemble des exemples de la classe « fissures » utilisée pour le jeu de test de la stratégie mono-grille de la cartographie par quadrillage régulier, à la différence que la vérité terrain y est ici relevée au pixel près. Le tableau 5.9 résume la composition de ces différents jeux. Il en ressort que très peu de sous-images contiennent des fissures, puisque seules 6,30% d'entre elles présentent une telle anomalie.

Apprentissage	Validation	Test (tout venant)	Test (restreint)
2640	2640	14960	942

TABLEAU 5.9 – Composition des jeux de données LCMS utilisés pour la cartographie des fissures par segmentation sémantique (en nombre de sous-images).

Nous avons vu, dans le chapitre 2, que les relevés des fissures admettaient une largeur d'un seul pixel, là où ces mêmes anomalies présentent visuellement une largeur variable, allant d'un pixel à une dizaine. Pour mesurer plus fidèlement les performances des modèles, il peut être utile d'avoir une métrique qui tienne compte des imprécisions spatiales au sein de la vérité terrain. Pour ce faire, nous calculons, en plus des métriques pixelliques utilisées jusqu'à présent, le θF_1-score, introduit par Drouyer [130]. Le θF_1-score est la moyenne harmonique du θ-rappel et de la θ-précision, qui sont eux-mêmes définis par

$$\theta\text{-rappel} = \frac{\#\left(\{\delta_\theta(b=1)\} \cap \{g=1\}\right)}{\#\{g=1\}} \tag{5.14}$$

et

$$\theta\text{-précision} = \frac{\#\left(\{b=1\} \cap \{\delta_\theta(g=1)\}\right)}{\#\{b=1\}} \tag{5.15}$$

où b est la carte de segmentation prédite par le modèle, valant 1 aux endroits où une fissure est détectée et 0 sinon, g la vérité terrain et δ_θ l'opérateur de dilatation morphologique par un carré de côté θ pixels. Ainsi, un pixel représentant une fissure mais prédit par le modèle comme revêtement sain contribuerait positivement au θ-rappel dès lors que ce pixel est à une distance de Manhattan (*i.e.* distance associée à la norme 1) inférieure à θ d'une détection de fissure. Symétriquement, un pixel annoté comme revêtement sain mais prédit comme fissure par le modèle ne pénaliserait pas la θ-précision pourvu que ce pixel se situe, selon la distance de Manhattan, à moins de θ pixels d'une fissure. En plus de prendre en compte l'incertitude sur le tracé des fissures au sein des vérités terrain, ces deux métriques reflètent possiblement mieux l'intérêt opérationnel des modèles évalués. En effet, en dehors des applications métrologiques, dans lesquelles on viserait à déterminer l'ouverture et/ou la longueur des fissures avec une résolution millimétrique, réussir à détecter une fissure à quelques pixels près demeure suffisant. Compte tenu de la largeur visuelle des fissures considérées, nous calculons le θF_1-score pour θ parcourant les nombres impairs allant de 3 à 11 (Drouyer utilise $\theta = 5$ dans ses travaux, mais les fissures qu'il étudie sont majoritairement plus larges que celles présentes dans nos données).

5.2.2.2 Résultats

Généralités Le tableau 5.10 présente les résultats quantitatifs obtenus par cette approche. De façon analogue aux approches par classification, on observe que ce sont les configurations utilisant conjointement l'intensité et la carte de profondeur brute D_r ou ajustée D_a, ainsi que la configuration réduite à l'intensité seule, qui présentent les meilleurs scores. En outre, ces résultats montrent que l'ajout des composantes L ou B ne se traduit jamais par une amélioration significative des métriques mesurées sur les modèles évalués. Concernant l'influence du critère de sélection du modèle, on peut noter que, bien que le critère EP permette d'atteindre une exactitude pondérée de plus de 90% et un rappel du même ordre, la précision correspondante dépasse au mieux 1% sur le jeu de test « tout venant », indiquant qu'environ 99% des pixels prédits comme fissure représentent en réalité des revêtements sains. Les modèles sélectionnés par le critère F_1 atteignent, pour certains, 14% de précision sur ce même jeu mais le rappel et, dans une moindre mesure, l'exactitude pondérée, s'en voient pénalisés. Un constat similaire peut être dressé pour le jeu de test restreint. Par ailleurs, on observe que l'exactitude pondérée pour chaque configuration est systématiquement plus basse pour le jeu restreint que sur le jeu « tout venant ». Comme le rappel est, par construction, identique pour ces deux jeux, on en déduit que c'est l'autre terme contribuant à l'exactitude pondérée, à savoir la spécificité (non montrée dans le tableau de résultats), qui diminue. Ainsi, la proportion de fausses alarmes (*i.e.* le nombre de fausses alarmes rapporté à la quantité de pixels en jeu) est plus élevée au voisinage des fissures que sur les images entières.

	Critère de sélection EP				Critère de sélection F_1			
	EP	P	R	F_1	EP	P	R	F_1
D_a	90.37	00.41	85.19	00.81	82.94	07.39	66.06	13.30
$D_a + B$	90.92	00.32	87.73	00.63	81.28	10.50	62.67	17.99
D_c	84.15	00.27	74.22	00.53	81.33	00.60	64.94	01.20
$D_c + L$	81.90	00.12	77.06	00.25	68.83	00.47	39.45	00.92
D_r	87.81	00.25	82.76	00.49	75.32	02.47	51.06	04.71
$D_r + L$	87.35	00.22	82.58	00.44	74.81	01.31	50.43	02.55
I	92.73	00.64	88.39	01.27	86.22	04.90	72.74	09.18
$I + D_a$	**94.96**	**01.10**	91.68	**02.17**	**88.15**	13.74	**76.41**	23.29
$I + D_a + B$	87.06	00.16	85.39	00.32	77.26	03.02	54.89	05.73
$I + D_c$	90.16	00.67	82.95	01.32	86.73	01.89	74.28	03.68
$I + D_c + L$	89.02	00.18	88.59	00.36	85.60	01.75	72.07	03.41
$I + D_r$	94.91	00.80	**92.25**	01.59	86.63	**14.22**	73.35	**23.82**
$I + D_r + L$	94.67	01.04	91.20	02.05	86.25	12.93	72.61	21.95

(a) Tout venant

	Critère de sélection EP				Critère de sélection F_1			
	EP	P	R	F_1	EP	P	R	F_1
D_a	88.71	03.59	85.19	06.88	82.27	12.86	66.06	21.53
$D_a + B$	87.11	02.15	87.73	04.20	80.70	14.40	62.67	23.42
D_c	83.41	03.29	74.22	06.29	80.65	05.70	64.94	10.49
$D_c + L$	80.62	01.62	77.06	03.18	68.34	04.60	39.45	08.23
D_r	85.43	02.30	82.76	04.48	74.86	11.43	51.06	18.68
$D_r + L$	85.84	02.50	82.58	04.86	74.37	09.22	50.43	15.58
I	91.60	05.46	88.39	10.28	85.78	17.35	72.74	28.02
$I + D_a$	93.00	05.19	91.68	09.83	**87.56**	16.67	**76.41**	27.37
$I + D_a + B$	85.70	02.03	85.39	03.96	76.84	13.24	54.89	21.33
$I + D_c$	89.22	05.86	82.95	10.95	86.20	11.79	74.28	20.35
$I + D_c + L$	87.83	02.27	88.59	04.42	85.11	11.64	72.07	20.04
$I + D_r$	92.92	04.65	**92.25**	08.85	86.17	**19.85**	73.35	**31.25**
$I + D_r + L$	**93.11**	**05.85**	91.20	**11.00**	85.75	18.14	72.61	29.03

(b) Jeu de test restreint

TABLEAU 5.10 – Scores obtenus (en %) sur le jeu de test par les différentes configurations pour la segmentation sémantique de sous-images LCMS (EP : exactitude pondérée ; P : précision ; R : rappel ; F_1 : F_1-score).

Analyse du θF_1-score La figure 5.17 montre, pour chaque configuration, l'évolution du θF_1-score, sur les deux jeux de test, lorsque θ varie de 1 à 11. Précisons que le θ-rappel est identique pour les jeux de test restreint et « tout venant ». Ainsi, des deux termes composant le θF_1-score, seule la θ-précision varie entre ces deux jeux. Notons, de plus, que, lorsque $\theta = 1$, le θF_1-score est égal au F_1-score.

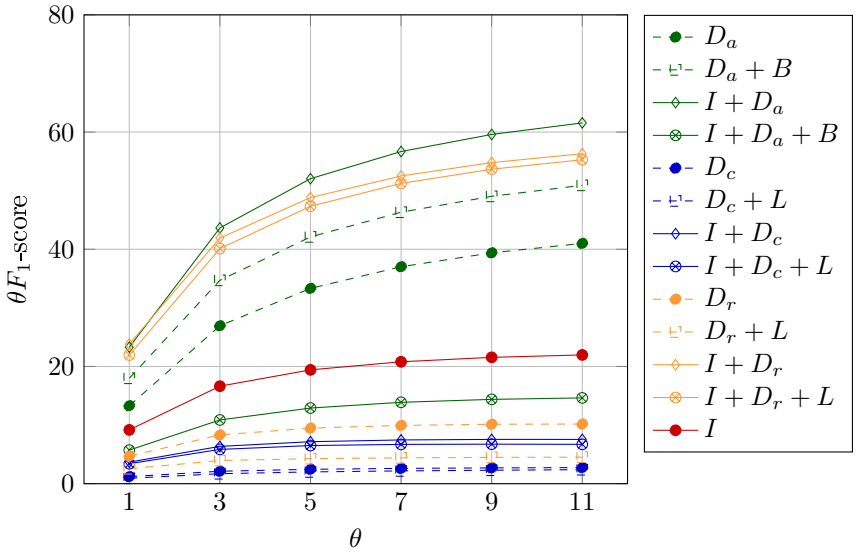
De façon générale, il ressort de ces résultats que les θF_1-scores les plus élevés sont atteints par des configurations employant l'information d'intensité. Par ailleurs, on constate que les configurations utilisant la profondeur centrée ainsi que $D_r + L$ admettent des F_1-scores significativement inférieurs aux autres configurations. De plus, on peut relever que l'ajout de la composante L au sein des configurations entraîne systématiquement une baisse du θF_1-score, et

ce pour toutes les valeurs θ considérées et pour les deux jeux de test. Une conclusion proche peut être dressée pour B. En effet, même si la configuration $D_a + B$ témoigne d'un θF_1-score plus élevé que celui de la configuration D_a, cet avantage disparaît lorsque la composante d'intensité est employée. Ainsi, la configuration $I + D_a$ présente un θF_1-score largement supérieur à celui obtenu par la configuration $I + D_a + B$.

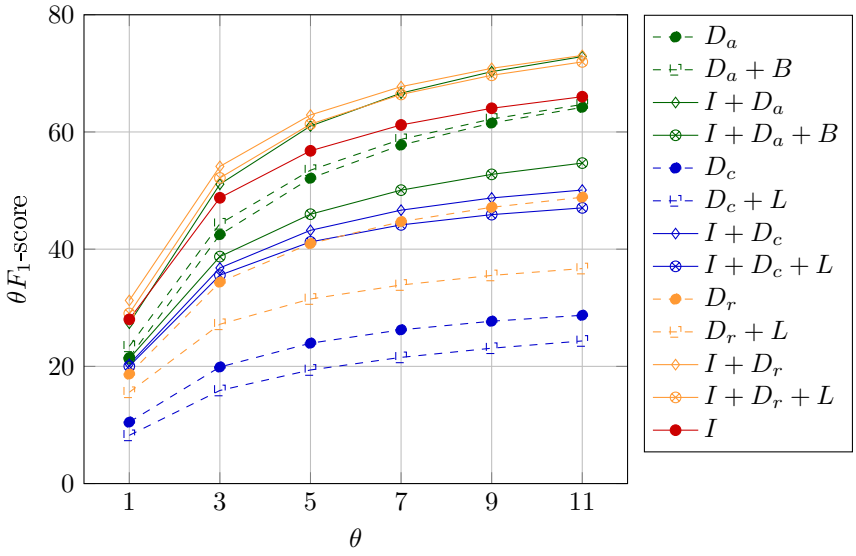
Concernant le jeu de test restreint, on peut relever qu'au fur et à mesure que θ grandit, le score de la configuration I, initialement très proche des configurations $I + D_a$, $I + D_r$ et $I + D_r + L$, s'éloigne des scores obtenus par ces dernières qui, quant à elles, se rejoignent. Cela signifie que les prédictions entre ces trois dernières configurations diffèrent principalement aux abords des fissures. Elles sont donc qualitativement proches. Par rapport à I, ces mêmes prédictions s'en distinguent également en d'autres endroits. En outre, les résultats obtenus montrent que la configuration $I + D_a$, dont le θF_1-score sur le jeu de test restreint est semblable à celui de la configuration $I + D_r$, le supplante nettement sur le jeu de test « tout venant ». Cela révèle que les prédictions du modèle issu de la configuration $I + D_r$ sont légèrement meilleures au voisinage des fissures que celles du modèle relatif à $I + D_a$. Elles présentent, à l'inverse, davantage de fausses alarmes que ces dernières au sein des sous-images qui ne comprennent pas ces anomalies.

Analyse qualitative En comparant les prédictions réalisées par les modèles sélectionnés par les deux critères (EP et F_1-score) sur un même exemple pour la configuration I (voir figure 5.18), il apparaît que le modèle sélectionné par le critère EP considère, à tort, toutes les aspérités du béton, ainsi que les joints et les fixations du passe-câble, comme des fissures. De plus, les bonnes détections de fissures « bavent » énormément, occupant par endroit une vingtaine de pixels de large quand les annotations des fissures admettent une largeur d'un seul pixel. Le critère F_1 permet de réduire ce trop grand nombre de détections. Ainsi, avec ce dernier, même si les deux fissures ne sont pas intégralement prédites, le nombre de fausses alarmes reste faible et ces dernières se concentrent majoritairement sur des zones difficiles (éraflures, joints localement non rectilignes).

Dans la figure 5.19, on compare qualitativement les prédictions réalisées pour les configurations I et $I + D_a$ afin d'étudier l'apport de la composante de profondeur ajustée, qui est la configuration qui présente le meilleur θF_1-score sur le jeu de test « tout venant ». Sur cet exemple, on peut voir que les fausses alarmes sont moins nombreuses et que le relevé des fissures est moins discontinu pour la configuration $I + D_a$. Pour réaliser la comparaison entre les prédictions de ces deux configurations, nous considérons trois zones d'intérêts, encadrées en vert dans la figure 5.19a, représentant respectivement une fissure excentrée (extrait de gauche), un joint de coffrage et un joint de voussoir (extrait central) et une fissure centrée traversée par un passe-câble ainsi qu'un joint de voussoir (extrait de droite). Il en ressort que le modèle appris sur l'intensité seule prédit régulièrement des portions de joints comme



(a) Tout venant



(b) Jeu restreint

FIGURE 5.17 – Évolution du θF_1-score (exprimé en %) sur le jeu de test pour les différentes configurations LCMS lorsque θ varie de 1 à 11 sur les modèles maximisant le critère F_1. Les courbes en pointillé représentent les configurations n'employant pas la composante d'intensité.

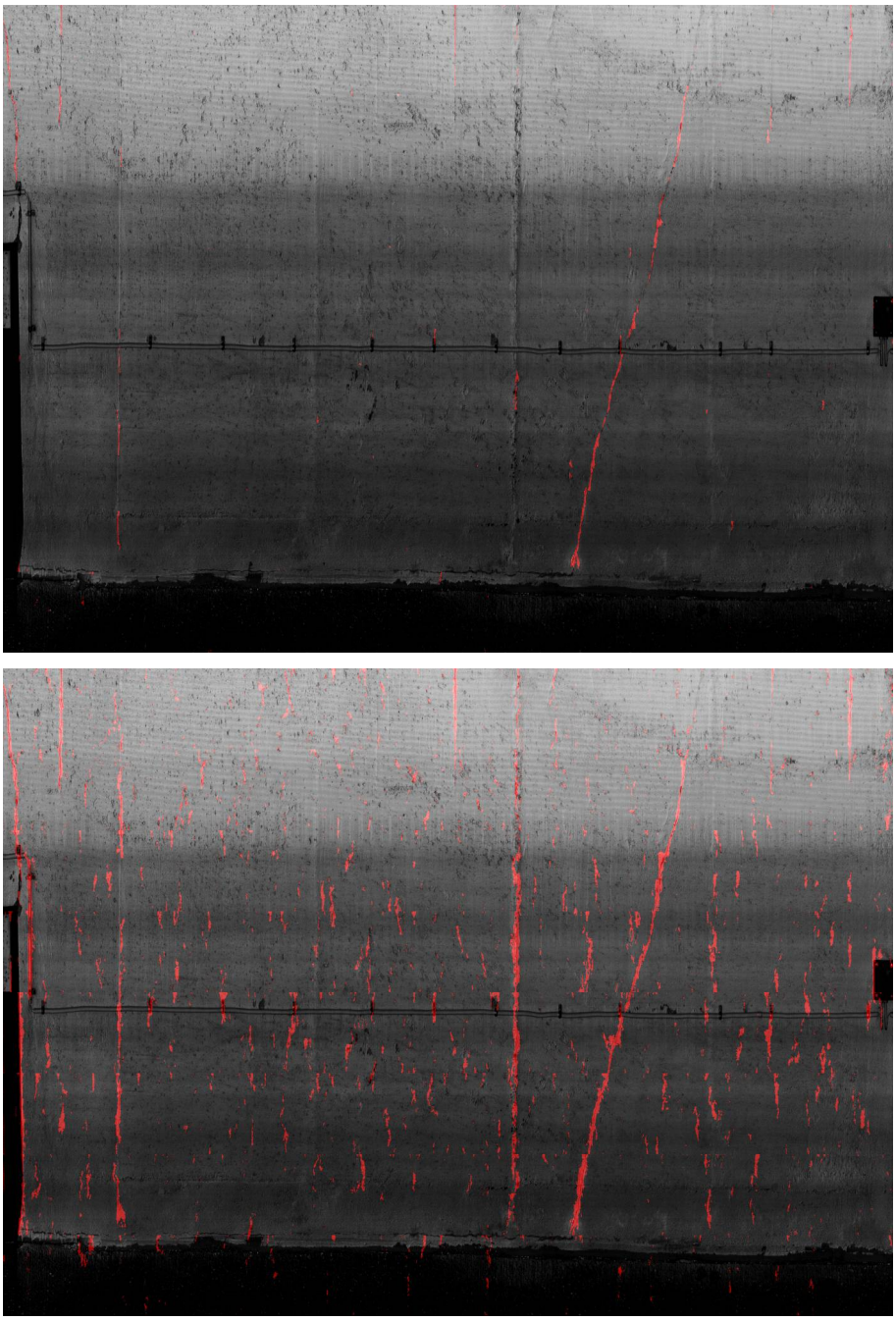


FIGURE 5.18 – Exemples de prédictions réalisées sur la même image de test que la figure 5.15 pour la segmentation sémantique et pour la configuration I. La sélection du modèle est fondée sur le F_1-score pour la prédiction du haut et sur l'exactitude pondérée (EP) pour celle du bas.

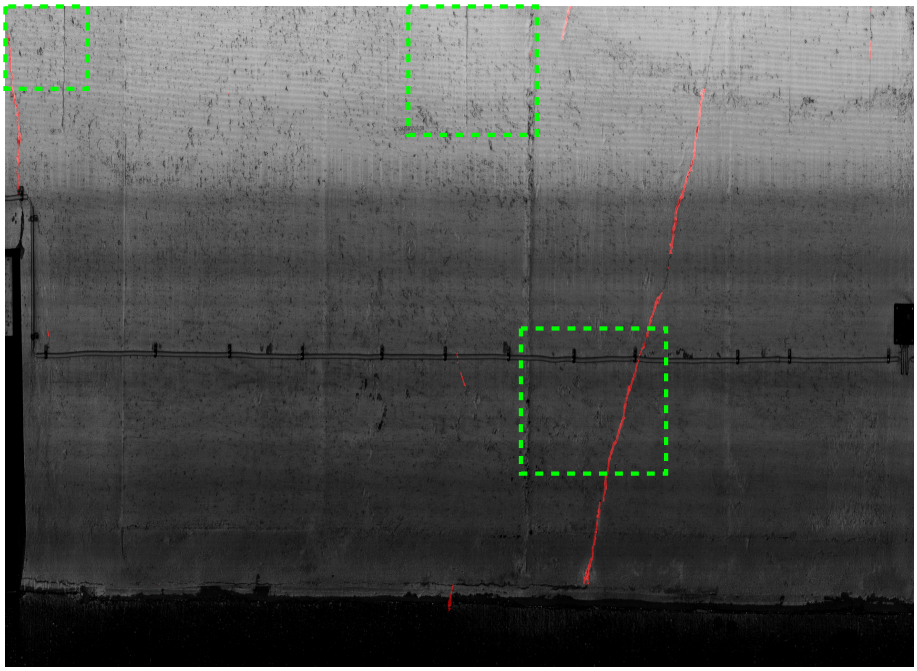
fissure, et ce quel que soit la largeur du joint. Une des fixations du passe-câble a également été prédite à tort comme fissure. Si les fissures sont globalement bien détectées, les relevés correspondants sont ponctués de discontinuités. À l'opposé, avec le modèle utilisant la profondeur ajustée en plus de l'intensité, les joints ne comptent presque jamais parmi les fausses alarmes. Concernant les fissures, elles sont un peu mieux prédites et la continuité de leurs relevés est davantage préservée.

Synthèse Ces travaux en segmentation sémantique viennent corroborer la plupart des conjectures émises lors de l'expérimentation sur la classification de données LCMS.

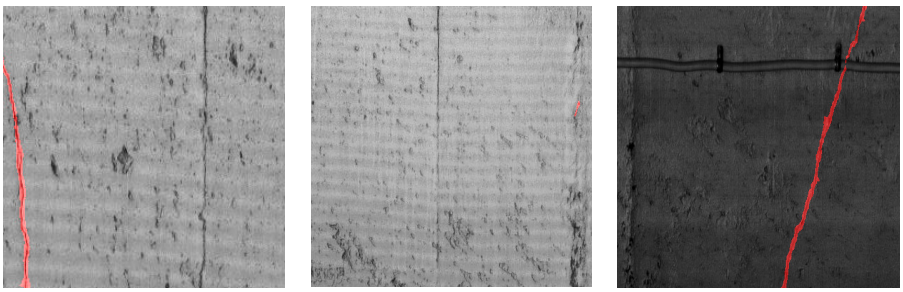
De façon attendue, l'écart entre les précisions obtenues par les modèles sélectionnés par chacun des deux critères (EP et F_1-score) est encore plus conséquent dans cette expérimentation que dans la précédente. Le même constat peut être dressé pour le rappel.

Concernant l'influence des différentes modalités, si l'intensité demeure indispensable, la composante de profondeur permet d'améliorer les résultats des modèles qui ne reposent que sur cette première modalité. Notons que, de prime abord, cela n'a rien d'évident. En effet, même si les fissures se traduisent matériellement par une variation de la surface du tunnel, l'intégralité des annotations a été réalisée sur la seule base de la carte d'intensité. On observe, plus spécifiquement, que l'information de profondeur ajustée D_a permet de confiner l'essentiel des fausses alarmes au voisinage des fissures. À l'opposé, certaines modalités ne semblent pas avoir d'influence positive sur la qualité des prédictions. C'est le cas de la carte des outliers B, de la carte des points hors de portée L et de la profondeur centrée D_c.

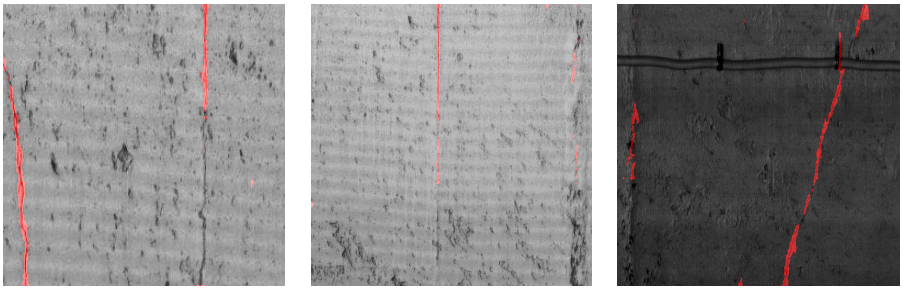
Enfin, il convient de souligner que, contrairement à l'expérimentation précédente, portant sur la cartographie par quadrillage régulier, nous n'avons rencontré aucun problème d'instabilité du processus d'optimisation lors de l'apprentissage des modèles.



(a) Prédiction obtenue pour la configuration $I + D_a$, critère de sélection F_1



(b) Extraits pour $I + D_a$, critère de sélection F_1



(c) Extraits pour I, critère de sélection F_1 (la prédiction entière est en figure 5.18)

FIGURE 5.19 – Comparaison qualitative des prédictions entre les configurations I et $I + D_a$ à partir de trois extraits (encadrés en vert) de prédictions issues d'un même exemple.

Conclusion du chapitre

Dans ce chapitre, nous avons évalué le potentiel des méthodes de cartographies par quadrillage régulier et par segmentation sémantique, sur deux types de données. Nous en tirons les conclusions suivantes :

— L'utilisation de jeux d'apprentissage de taille réduite et/ou faiblement représentatifs de l'ouvrage à inspecter ne permet pas d'obtenir des réseaux de neurones généralisant bien sur l'intégralité de l'ouvrage en question. Dans cette configuration, les modèles employant des caractéristiques définies *a priori*, à l'instar des forêts aléatoires ou des *SVM*, semblent plus robustes à ce manque de représentativité, puisque les caractéristiques sur lesquelles repose la décision du modèle sont expressément choisies en raison de leur intérêt pour l'identification des anomalies, là où les réseaux de neurones infèrent ces caractéristiques sur les données d'apprentissage.

— À l'inverse, lorsqu'ils sont appris sur une portion significative des données représentant l'ouvrage à analyser, les réseaux de neurones présentent de bonnes capacités de généralisation sur les autres données de la structure, qui leur sont inédites.

- Pour la segmentation sémantique des fers apparents au sein d'images photographiques, nous sommes parvenus à atteindre une exactitude pondérée proche de 90% et un F_1-score supérieur à 50% et ce, pour l'ensemble des sites considérés.
- Concernant la reconnaissance des fissures par le biais des données LCMS, les résultats varient selon les modalités des données choisies ainsi que de la stratégie de cartographie employée. Néanmoins, l'exactitude pondérée des meilleures configurations s'établit également autour de 90%, que ce soit pour la cartographie par quadrillage régulier ou par segmentation sémantique. S'agissant du F_1-score, il dépasse 65% pour la configuration D_a et la stratégie multi-grilles (approche Cha *et al.* [9]) et le θF_1-score avoisine 62% pour la configuration $I + D_a$ et la stratégie par segmentation sémantique (lorsque θ vaut 11).

— Pour le cas particulier des images fortement résolues, ou qui présentent des infrastructures acquises en plan rapproché, il est possible d'améliorer les performances d'un modèle de segmentation sémantique en combinant, pour chaque image à cartographier, les prédictions obtenues sur plusieurs versions sous-échantillonnées de cette image.

— Lorsque la classe d'intérêt est largement minoritaire, les deux critères de sélection évalués proposent des compromis précision/rap-

pel différents : la sélection du modèle selon le F_1-score favorise la précision sur le rappel alors que la sélection selon l'exactitude pondérée permet d'atteindre un rappel plus élevé, au prix d'un nombre plus important de fausses alarmes.

— Certaines modalités des données LCMS ont davantage d'influence sur les performances des modèles que d'autres et ce, aussi bien pour la cartographie par quadrillage régulier que pour celle reposant sur la segmentation sémantique. Ainsi :

- La composante d'intensité est essentielle. En son absence, les résultats ont globalement tendance à être moins bons.
- L'utilisation conjointe de la profondeur (brute D_r, ou ajustée D_a) et de l'intensité représente la meilleure stratégie pour obtenir des modèles performants. Par ailleurs, il ressort de l'analyse du θF_1-score des modèles de segmentation sémantique que l'emploi de la profondeur ajustée permet au modèle de confiner l'essentiel des fausses alarmes dans le voisinage immédiat des fissures.
- Les autres composantes considérées (profondeur centrée D_c, carte des points hors de portée L et carte des *outliers* B) ne semblent pas avoir une incidence significative sur les performances des modèles.

Chapitre 6

Évaluation multi-sites et stratégies d'adaptation de domaine pour la détection d'anomalies sur des structures de génie civil

Dans le chapitre précédent, nous avons évalué les performances des réseaux de neurones sur des données issues du même ouvrage que celui ayant servi à l'apprentissage. Si cette évaluation est un point d'étape essentiel, elle demeure insuffisante pour se prononcer sur la capacité de généralisation des modèles, c'est-à-dire lorsque des images provenant d'un tunnel nouvellement acquis sont à analyser.

Ce chapitre a pour but de mesurer l'influence du biais de domaine dans le cadre de la segmentation sémantique des fers apparents ainsi que d'évaluer le potentiel de diverses approches d'adaptation de domaine. Sur le plan opérationnel, on veut éviter d'avoir à annoter de larges portions de chaque tunnel à cartographier. Le temps nécessaire à cette opération serait considérable et l'intérêt des méthodes de cartographie automatiques s'en verrait réduit.

Sommaire

6.1 Évaluation de l'influence du biais de domaine

Résumé du protocole expérimental

- **Architecture** SegNet
- **Poids initiaux** Pré-appris (encodeur), aléatoires (décodeur)
- **Époques** 1000
- **Optimiseur** *Adam*
- **Augmentation de données** Drouyer [130]
- **Critère d'optimisation** Entropie croisée pondérée
- **Critères de sélection des modèles à évaluer**
  - Modèle maximisant le F_1-score sur le jeu de validation
  - Modèle maximisant l'exactitude pondérée sur ce même jeu
- **Objectif**
  - Évaluer les performances des réseaux de neurones sur des jeux de données de grande taille et en présence d'un important biais de domaine

6.1.1 Méthodologie

Pour mesurer la capacité d'adaptation de nos réseaux, il nous faut entraîner un modèle sur un site et l'évaluer sur un autre. Nous disposons de jeux annotés pour cinq sites d'acquisitions. Cependant, le nombre de jeux au sein de chaque site n'est pas constant. CODEBRIM est la seule base composée de trois jeux : apprentissage validation et test. Les tunnels piéton et routier (Rive-de-Gier) sont tous deux constitués d'un jeu d'apprentissage et d'un jeu de test. Les deux acquisitions sur le bâtiment universitaire ne comptabilisent chacune qu'un seul jeu.

Comme l'évaluation porte à chaque fois sur un site distinct du site d'apprentissage, les jeux initialement dédiés aux tests du tunnel piéton et du tunnel routier de Rive-de-Gier sont réemployés en tant que jeux de validation et servent ainsi à la sélection du modèle à tester selon les deux critères que nous avons employés jusqu'à présent : sélection du modèle réalisant la meilleure exactitude pondérée et sélection de celui ayant obtenu le meilleur F_1-score en validation. Pour la base CODEBRIM, qui dispose déjà d'un jeu de validation, ce dernier est conservé pour ce rôle. Les jeux du bâtiment universitaire ne sont utilisés qu'en test. Le tableau 6.1 présente les différentes configurations testées.

Apprentissage et validation	Test
Tunnel piéton (TP)	C, RDG, Bâtiment universitaire (A et S)
CODEBRIM (C)	TP, RDG, Bâtiment universitaire (A et S)
Rive-de-Gier (RDG)	C, TP, Bâtiment universitaire (A et S)

Tableau 6.1 – Configurations évaluées pour mesurer l'influence du biais de domaine (A : captation par l'objectif Apple ; S : captation par l'objectif Samsung).

Nous reprenons les modèles appris lors de l'expérimentation portant sur la segmentation sémantique des fers apparents décrite dans le chapitre 5. Chaque modèle a donc été évalué sur le jeu de validation qui lui est associé et les critères de sélection ont alors désigné, parmi l'ensemble des modèles générés lors de la phase d'apprentissage, ceux qui sont à évaluer sur les différents jeux de test. L'apprentissage et la validation sont réalisés sur des sous-images 256×256 pixels extraites selon les facteurs d'échelle propres à chaque site :

— $\{2^{-2}, 2^{-3}\}$ pour le tunnel piéton ;

— $\{2^{-1}\}$ pour le tunnel de Rive-de-Gier ;

— $\{2^{-2}\}$ pour la base CODEBRIM.

L'évaluation porte sur les images en pleine résolution. Le tableau 6.2 présente la composition des jeux de données pour chaque site.

Site	Apprentissage (#s.i.)	Validation (#s.i.)	Test
Tunnel piéton	1876	1092	78
CODEBRIM	3173	229	29
Rive-de-Gier	6060	2508	418
Rive-de-Gier (R)	–	–	27
Bât. univ. (A)	–	–	56
Bât. univ. (S)	–	–	17

TABLEAU 6.2 – Composition des jeux de données pour les différents sites dans le cadre de l'évaluation du biais de domaine. Les jeux d'apprentissage et de validation sont comptés en sous-images 256×256 pixels, les jeux de test le sont en images, considérées dans leurs résolutions originales. Pour le tunnel piéton et Rive-de-Gier, les jeux de validation et de test sont composés des mêmes images (ces jeux ne sont jamais utilisés simultanément pour un même modèle). (R) : jeu de test restreint de Rive-de-Gier.

6.1.2 Résultats

Les scores des différents modèles sont présentés dans le tableau 6.3. Nous les analysons séparément avant d'en faire une synthèse générale. À l'exception du bâtiment universitaire, toutes les images de test ont déjà été utilisées et décrites dans le chapitre 5 (section 5.1.2). Pour le bâtiment universitaire, nous avons sélectionné une image de chacune des deux sous-bases. L'ensemble de ces exemples est présenté en figure 6.1.

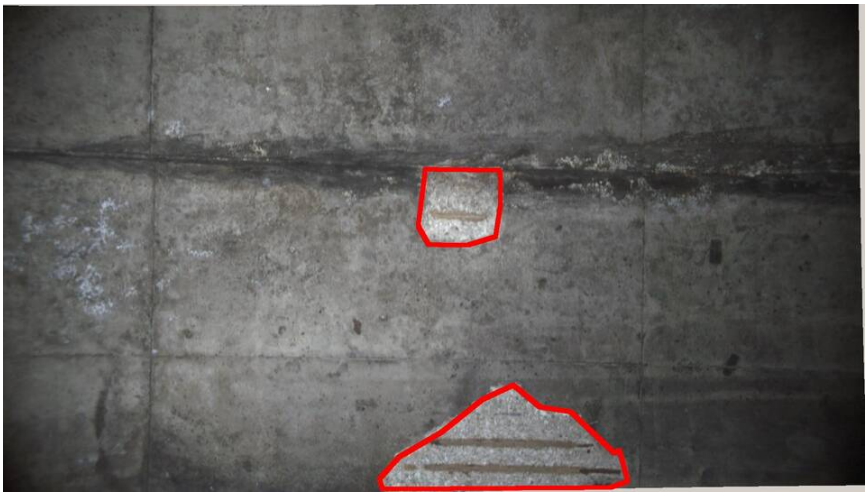
Apprentissage sur le tunnel piéton Lorsque le modèle est appris sur le tunnel piéton (*cf.* table 6.3a), on mesure, sur le tunnel Rive-de-Gier, une exactitude pondérée proche de 50% ainsi qu'un F_1-score quasi-nul. Les prédictions sur ce tunnel (voir figure 6.2) confirment ce résultat puisqu'on y constate simultanément une quasi absence de détection et une quantité importante de fausses alarmes. On observe des résultats analogues pour le rappel par composantes connexes (*cf.* figure 6.3) : lorsque l'on se base sur le critère du F_1-score,

(a) Bâtiment universitaire (jeu A)

(b) Bâtiment universitaire (jeu S)

(c) Tunnel piéton

(d) CODEBRIM



(e) Tunnel routier de Rive-de-Gier

FIGURE 6.1 – Exemples utilisés pour l'évaluation qualitative des différents modèles. Les fers apparents sont détourés en rouge.

	Critère EP				Critère F_1			
	EP	P	R	F_1	EP	P	R	F_1
Rive-de-Gier	52.65	00.42	06.81	00.78	50.18	00.20	00.68	00.30
Rive-de-Gier (R)	52.63	06.11	06.81	06.44	50.19	03.29	00.68	01.13
CODEBRIM	52.99	82.83	06.43	11.93	55.98	85.53	12.67	22.07
Bât. univ. (S)	71.73	04.44	50.40	08.15	78.43	08.09	61.33	14.29
Bât. univ. (A)	71.10	27.12	43.90	33.53	80.40	15.14	66.11	24.64

(a) Modèle appris sur le tunnel piéton

	Critère EP				Critère F_1			
	EP	P	R	F_1	EP	P	R	F_1
Tunnel piéton	44.51	01.53	13.54	02.76	50.00	02.92	00.05	00.10
CODEBRIM	53.49	62.62	08.69	15.27	50.10	41.70	00.36	00.71
Bât. univ. (S)	50.39	01.65	01.26	01.43	49.99	00.00	00.00	00.00
Bât. univ. (A)	42.83	00.48	07.37	00.91	49.98	00.27	00.01	00.02

(b) Modèle appris sur Rive-de-Gier

	EP	P	R	F_1
Rive-de-Gier	65.78	00.52	38.40	01.02
Rive-de-Gier (R)	65.09	06.50	38.40	11.11
Tunnel piéton	51.12	06.21	03.91	04.80
Bât. univ. (S)	73.77	09.84	50.49	16.46
Bât. univ. (A)	72.75	15.62	49.33	23.73

(c) Modèle appris sur la base CODEBRIM (les critères EP et F_1 aboutissent au même modèle)

TABLEAU 6.3 – Métriques obtenues pour les différentes configurations (EP : exactitude pondérée ; P : précision ; R : rappel ; F_1 : F_1-score) – (R) : évaluation sur le jeu de test restreint de Rive-de-Gier.

aucun fer apparent n'est détecté à plus de 10% sur Rive-de-Gier et moins de 10% des fers apparents présents dans ce même tunnel sont concernés par une détection. Avec le critère EP, le modèle de cette configuration ne réussit à détecter que 10% des composantes connexes au moins de moitié. Cette situation est probablement due à la différence d'aspect importante entre les deux sites. En effet, pour le tunnel de Rive-de-Gier, les pertes de matière entourant les fers apparents sont plus claires que le revêtement sain les avoisinant alors que la situation est inverse pour le tunnel piéton. De plus, sur l'image de test, on peut voir que le joint présent, visuellement composé d'une structure rectiligne entouré d'une zone granuleuse sombre, est partiellement détectée comme fer apparent.

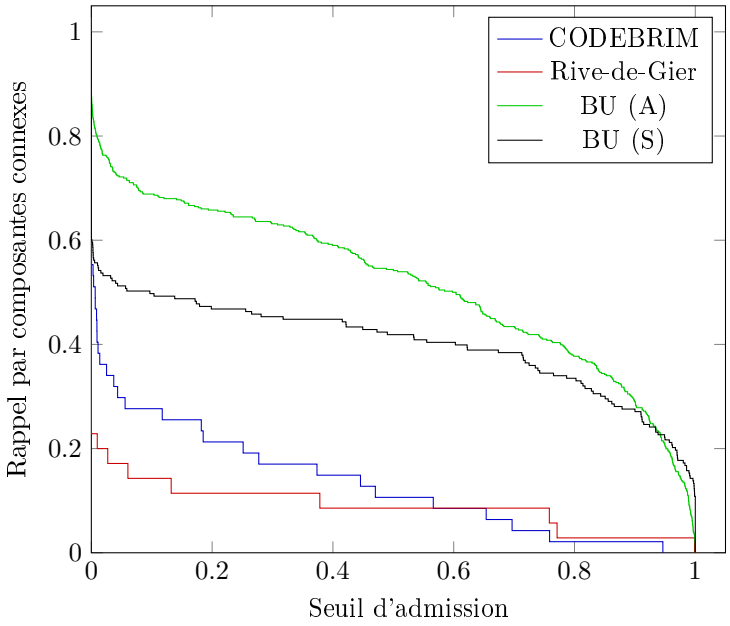
Concernant la base CODEBRIM, on retrouve une exactitude pondérée semblable à celle de Rive-de-Gier mais le F_1-score y est cette fois supérieur. Notons qu'une partie de ce score est expliquée par le fait que la plupart des fers apparents sont pris en plan rapproché et occupent donc une grande surface au

sein des images. La précision s'en voit ainsi favorablement biaisée. Cette faible distance entre le capteur et la paroi explique également le faible rappel par composantes connexes. En effet, puisque les anomalies sont de larges composantes connexes dans les images, il est plus difficile de les détecter intégralement. Ainsi, on dénombre moins de 20% de fers apparents détectés avec un seuil d'admission de 50%, et ce pour les deux critères. Ce constat se retrouve au sein des prédictions, où l'on observe que le modèle ne parvient qu'à localiser approximativement une partie des fers apparents, et ce, pour les deux critères. Remarquons que les fers apparents les moins bien détectés dans cette image de test sont ceux qui sont situés sur une arête de l'ouvrage. Or, une telle situation ne se retrouve pas au sein du tunnel piéton, pour lequel toutes les anomalies sont situées sur une surface planaire, alors qu'elle est courante pour la base CODEBRIM. La sous-détection du modèle appris sur le tunnel piéton pourrait donc, en partie, s'expliquer par cette différence de caractéristiques d'anomalie entre les deux sites.

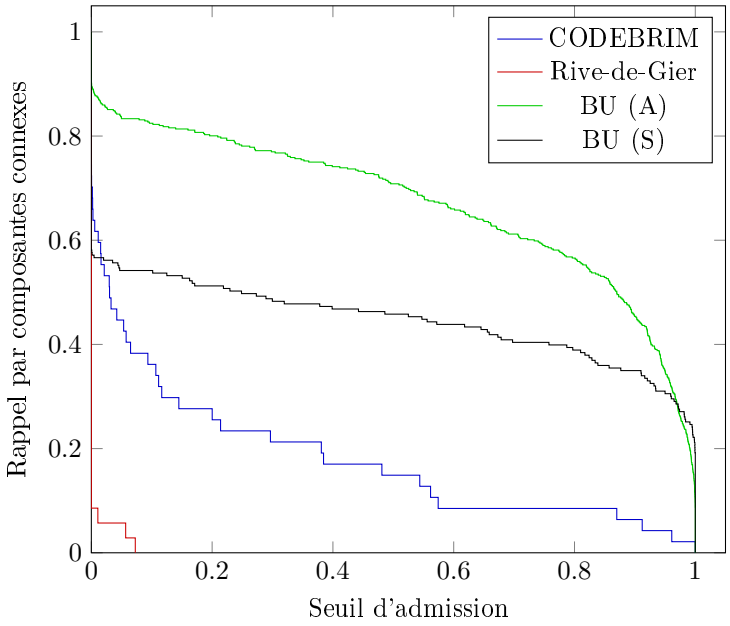
Pour le bâtiment universitaire, (S) comme (A), on observe de meilleurs scores, aussi bien en exactitude pondérée qu'en F_1-score. Néanmoins, si le rappel est plus élevé que sur les autres bases, attestant que la plupart des fers apparents y sont mieux détectés, la précision demeure faible. Ainsi, les exemples de prédictions révèlent que certains éléments distracteurs, comme les branches et les jonctions entre blocs de béton, sont classés à tort comme fers apparents. On peut expliquer cette contre-performance par l'absence de ces éléments dans le jeu d'apprentissage. En effet, le bâtiment universitaire comporte des éléments faiblement représentés au sein des tunnels, comme la végétation. Les prédictions incorrectes réalisées sur ces éléments rarement présents sont donc moins dommageables pour notre application. De façon générale, on peut noter que le rappel par composantes connexes est plus élevé pour (A) que pour (S). En effet, pour le critère EP, on détecte près de 50% des fers apparents avec un seuil d'admission de 50% pour (S) et, avec ce même seuil, ce sont un peu moins de 60% des anomalies qui sont détectées dans (A). Pour le critère F_1, on relève des rappels par composantes connexes du même ordre pour (S) et plus élevés pour (A), s'établissant respectivement à 50% pour le premier et à 75% pour le second, avec un même seuil d'admission à 50%. Cette différence tient sans doute en partie à la différence de netteté et de résolution des images. En effet, les fers apparents du jeu (A) sont davantage discernables que ceux du jeu (S).

Critère EP Critère F_1

FIGURE 6.2 – Prédictions du modèle appris sur le tunnel piéton – **1ère ligne :** Bâtiment universitaire (S) ; **2ème ligne :** Bâtiment universitaire (A) ; **3ème ligne :** CODEBRIM ; **4ème ligne :** Rive-de-Gier.



(a) Critère EP



(b) Critère F_1

FIGURE 6.3 – Rappel par composantes connexes pour le modèle appris sur le tunnel piéton.

Apprentissage sur le tunnel de Rive-de-Gier Pour le modèle entraîné sur le tunnel routier de Rive-de-Gier (*cf.* table 6.3b et figure 6.4), on observe des métriques uniformément basses, s'établissant parfois largement en deçà des résultats qu'auraient produits des modèles dégénérés ou répondant au hasard. Avec le critère EP, le modèle génère de nombreuses fausses alarmes (végétation, fenêtres, toit, ciel, revêtements sains, etc.) et ne réussit qu'à détecter des fers apparents que sur quelques exemples de la base CODEBRIM et du bâtiment universitaire (S). En termes de rappel par composantes connexes (*cf.* figure 6.5), entre 10 et 20% des fers apparents sont détectés avec un seuil d'admission de 50% (à l'exception du bâtiment universitaire (S) qui présente un rappel par composantes connexes de 0,49% pour ce même seuil d'admission). Avec le critère F_1, les prédictions sont presque dégénérées et ne comportent que peu de détections. Cela se traduit, pour le rappel par composantes connexes, par le fait qu'aucune d'entre elles n'est reconnue à plus de 20%. Cette très faible capacité de généralisation pourrait s'expliquer par une importante hétérogénéité entre le tunnel de Rive-de-Gier et l'ensemble des autres sites. En effet, dans le chapitre introductif, nous avions réalisé une représentation t-SNE de l'ensemble des images de chaque site (*cf.* figure 1.9) et avions pu constater que le tunnel de Rive-de-Gier présentait une « distribution » très éloignée des autres sites qui admettaient, quant à eux, des représentations davantage rapprochées.

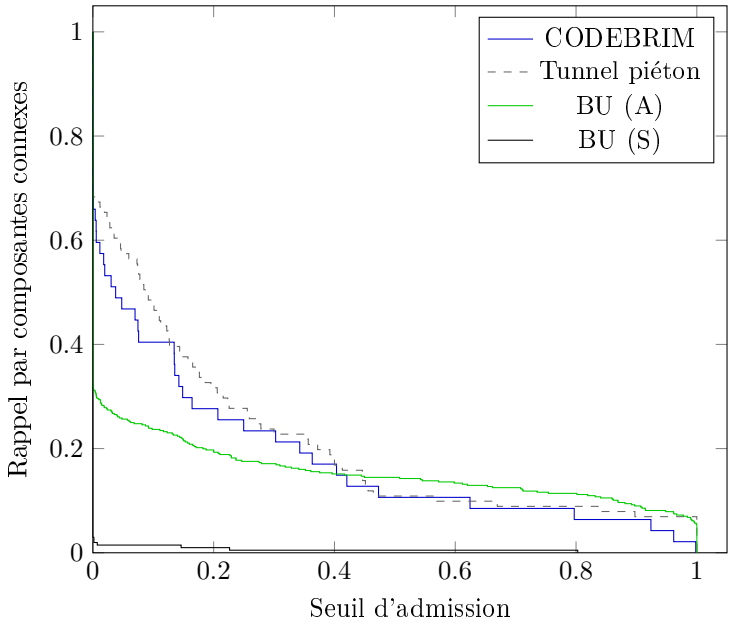
Apprentissage sur la base CODEBRIM L'apprentissage effectué sur la base CODEBRIM nous permet d'obtenir un modèle généralisant légèrement mieux que ceux appris sur les deux autres sites (*cf.* tableau 6.3c et figure 6.6). On retrouve des résultats comparables aux modèles appris sur le tunnel piéton pour les deux versions du bâtiment universitaire, aussi bien sur le plan quantitatif que qualitatif. Les précisions mesurées sur l'ensemble des sites indiquent cependant une quantité importante de fausses alarmes.

Évalué sur le tunnel piéton, le modèle parvient à détecter partiellement les fers apparents dans l'exemple choisi mais, considère, à tort, une partie du revêtement composant le sol comme anomalie, probablement en raison des granulats qu'il contient et qui lui donnent une apparence proche des pertes de matières entourant les fers apparents. En termes de rappel par composantes connexes (*cf.* figure 6.7), on constate qu'aucune composante connexe n'est détectée à plus de 95% de sa superficie (seuil d'admission à 95%) et qu'au plus 42% des composantes connexes sont détectées (seuil à 0%). Ces détections sont donc fortement incomplètes. La cause de cette incomplétude est probablement à chercher du côté du champ réceptif du réseau de neurones. En effet, certaines prises de vue du tunnel piéton sont réalisées en plan rapproché. Comme les images sont fortement résolues (4896×3672), le modèle ne peut possiblement pas prendre en compte suffisamment de contexte spatial pour identifier certains fers apparents pris en gros plan.

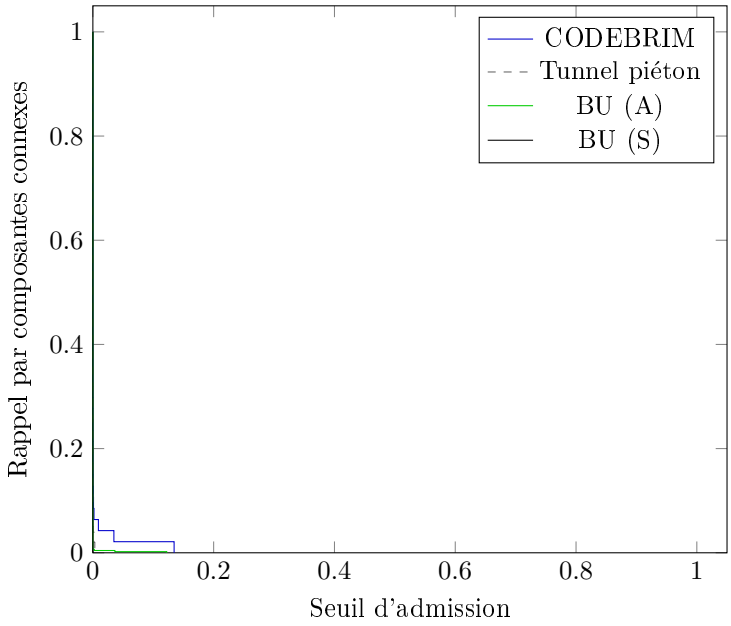
Sur le tunnel routier, les résultats sont meilleurs que ceux du modèle appris sur le tunnel piéton. En effet, si les deux modèles témoignent de précisions proches, celui entraîné sur la base CODEBRIM admet un rappel de 38.40%,

Critère EP Critère F_1

FIGURE 6.4 – Prédictions du modèle appris sur le tunnel routier de Rive-de-Gier – **1ère ligne :** Bâtiment universitaire (S) ; **2ème ligne :** Bâtiment universitaire (A) ; **3ème ligne :** CODEBRIM ; **4ème ligne :** Tunnel piéton.



(a) Critère EP



(b) Critère F_1

FIGURE 6.5 – Rappel par composantes connexes pour le modèle appris sur le tunnel routier de Rive-de-Gier.

(a) Bâtiment universitaire (S)

(b) Bâtiment universitaire (A)

(c) Tunnel piéton

(d) Rive-de-Gier

FIGURE 6.6 – Prédictions du modèle appris sur la base CODEBRIM (les critères EP et F_1 aboutissent au même modèle).

contre 6.81% pour le modèle issu du tunnel piéton. Le premier modèle détecte donc davantage de fers apparents que le second, constat qui est confirmé qualitativement sur la prédiction (présentée en figure 6.6d) ainsi qu'au niveau des composantes connexes. En effet, pour tous les seuils d'admission, le rappel par composantes connexes du modèle appris sur la base CODEBRIM est supérieur à ceux des deux modèles entraînés sur le tunnel piéton.

Concernant le bâtiment universitaire, on retrouve la même confusion du modèle au niveau des branches, peu représentées dans la base CODEBRIM, qui sont considérées à tort comme fers apparents, pour (S) comme pour (A). De plus, dans le jeu (A), le brise-soleil ainsi que l'ombre qu'il projette sur les vitres du bâtiment comptent également parmi les fausses alarmes. On peut noter que la base CODEBRIM compte de nombreux fers apparents situés sur des arêtes de structures en béton. La présence de ces derniers a pu conduire le modèle à prédire le brise-soleil comme fer apparent. Bien que le rappel pixellique soit semblable pour les deux jeux, le rappel par composantes connexes diffère significativement pour chacun d'eux. Ainsi, pour tous les seuils d'admission inférieurs à 90%, le rappel par composantes connexes est inférieur sur (S) à ce qu'il est sur (A), l'écart se creusant lorsque le seuil d'admission décroît. Les détections sur (A) enveloppent ainsi davantage les fers apparents que celles réalisées sur (S).

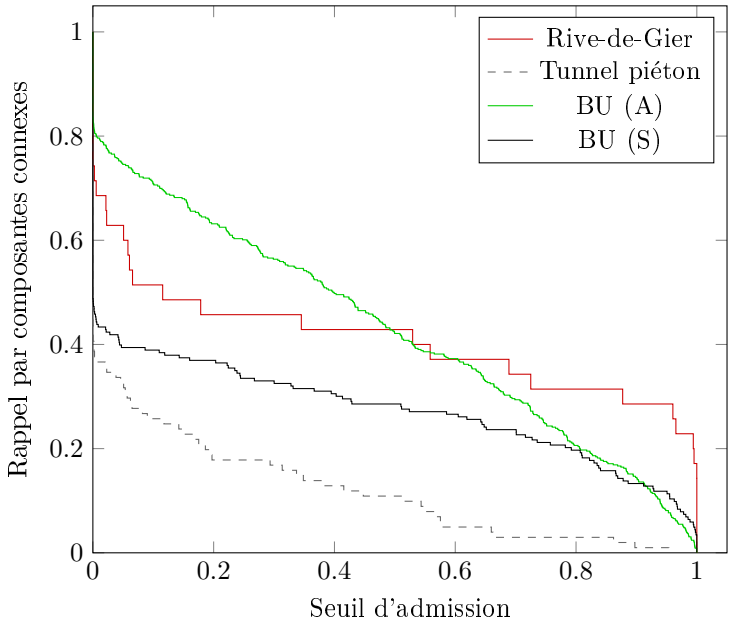


FIGURE 6.7 – Rappel par composantes connexes pour le modèle appris sur la base CODEBRIM.

6.1.3 Synthèse

Dans cette section, nous avons pu mesurer l'influence du biais de domaine pour la segmentation sémantique des fers apparents. Comme l'on pouvait s'y attendre, il en ressort une différence significative de performances entre une évaluation sur le même site que celui ayant servi à son apprentissage et une évaluation sur un autre site, situation qui se rapproche davantage de l'objectif applicatif de la cartographie des anomalies.

Il nous faut alors gérer explicitement le biais de domaine au sein du modèle. Cette problématique est bien connue dans la communauté de la reconnaissance des formes et a donné lieu à des méthodes qualifiées d'« adaptation de domaine » qui font l'objet d'une littérature conséquente [146]. La plupart de ces méthodes proposent de traiter le problème de biais de domaine de façon binaire, c'est-à-dire de ne considérer le biais qu'entre deux domaines, le procédé étant alors à réitérer pour chaque domaine supplémentaire. C'est cette stratégie que nous avons choisie. Le premier jeu est qualifié de domaine source et est nécessairement annoté tandis que le second est appelé domaine cible et ne l'est généralement pas (ou alors partiellement) lors de la phase d'apprentissage.

Bien que les résultats de cette étude multi-sites n'aient pas fait l'objet de valorisation scientifique, certains travaux préliminaires mesurant de l'influence du biais de domaine, pour une architecture U-Net [5], ont été publiés en conférences nationale [147] et internationale [145].

6.2 Adaptation de domaine

Nous explorons et comparons deux approches d'adaptation de domaine : une approche non supervisée (section 6.2.1), d'une part, et une approche faiblement supervisée (section 6.2.2), d'autre part. Bien que ce ne soit pas une limitation scientifique, nous restreignons l'expérimentation d'adaptation de domaine au site de Rive-de-Gier comme domaine cible. C'est, en effet, le site pour lequel la variabilité d'aspect des fers apparents (échelle, teinte, etc.) est la plus faible. Ainsi, on peut espérer bénéficier d'un effet « seuil » : si on parvient à correctement détecter un fer apparent, il est probable que les autres soient également bien reconnus.

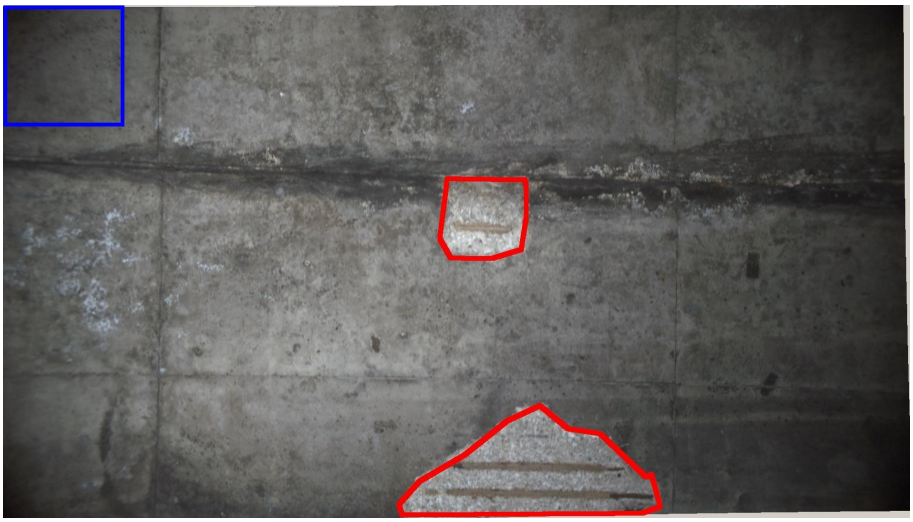
Pour estimer la qualité de l'adaptation de domaine par-delà les indicateurs mesurés sur le domaine cible, il serait intéressant de vérifier si les caractéristiques produites par le modèle sur ces deux domaines suivent la même distribution statistique. S'il est complexe de tester avec précision cette affirmation, on peut toutefois s'approcher de cet objectif à travers une visualisation t-SNE [1] des caractéristiques issues de l'encodeur. S'intéresser à la distribution en sortie de l'encodeur est motivé par plusieurs raisons. Une raison méthodologique, tout d'abord, puisqu'une des deux méthodes d'adaptation que nous évaluons n'agit que sur l'encodeur. Une raison technique, ensuite, puisque c'est à cet endroit du réseau que le flux d'information est le moins volumineux. Il est donc possible d'appliquer l'algorithme de visualisation directement sur les données brutes, sans réduction de dimension préalables (pour les données trop volumineuses pour une application directe de t-SNE, les auteurs de cette méthode préconisent d'effectuer une analyse par composantes principales en guise de prétraitement, ce qui peut dégrader la pertinence de la visualisation). Le résultat de l'encodeur, étant donnée une image RGB de résolution 256×256 pixels, est un volume $8 \times 8 \times 512$. Dans cette représentation, chaque point représente alors un « pixel » de ce volume, c'est-à-dire un volume $1 \times 1 \times 512$, et est associé à un voisinage 32×32 de l'image originale.

Compte tenu de la taille des jeux de données, il serait déraisonnable de calculer la représentation t-SNE pour toutes les images. Il en résulterait des nuages de points de grandes densités, inégalement peuplés, et donc difficilement interprétables. Pour Rive-de-Gier, nous sélectionnons ainsi une unique image tandis que pour le tunnel piéton et la base CODEBRIM, nous en choisissons deux que nous redimensionnons d'un facteur 1/4, de sorte que le nombre de pixels considérés pour la représentation t-SNE soit approximativement le même pour les trois domaines. Ces deux couples d'images ont été composés d'exemples visuellement semblables, ceci afin que chaque paire d'images ainsi formée se comporte, au sein de la visualisation, de la même façon qu'une seule image de résolution supérieure.

L'ensemble des images utilisées pour la production des représentations t-SNE (ainsi que les facteurs d'échelle qui leur sont associés) est donné en figure 6.8.

(a) Exemples issus du tunnel piéton (pris à l'échelle 1/4)

(b) Exemples issus de la base CODEBRIM (pris à l'échelle 1/4)



(c) Exemple issu du tunnel de Rive-de-Gier (pris à l'échelle 1)

FIGURE 6.8 – Exemples utilisés pour la visualisation t-SNE des caractéristiques produites par l'encodeur des modèles évalués. Un facteur d'échelle est appliqué sur les exemples du tunnel piéton et de la base CODEBRIM afin que le nombre de pixels soit du même ordre de grandeur au sein de chacun des domaines. Pour donner une idée des résolutions originales des images, un carré bleu de taille 256×256 est tracé dans le coin supérieur gauche de chaque image.

6.2.1 Méthode non supervisée

Résumé du protocole expérimental

- **Architecture** SegNet
- **Poids initiaux**

Phase autosupervisée :
– Pré-appris (encodeur), aléatoires (décodeur)
Phase supervisée :
– Modèle issu de la phase autosupervisée

- **Époques** 1000 + 1000
- **Optimiseur** *Adam*
- **Augmentation de données** Algorithme 3
- **Critères d'optimisation** $\mathcal{L}_{self}$ puis $\mathcal{L}_{ce}$
- **Critères de sélection des modèles à évaluer**

Phase supervisée uniquement :
– Modèle maximisant le F_1-score sur le jeu de validation du domaine source
– Modèle maximisant l'exactitude pondérée sur ce même jeu

- **Objectif**

– Évaluer une approche d'adaptation de domaine non supervisée

La première méthode mise en œuvre repose en partie sur un type d'apprentissage particulier appelé « autosupervision » [148].

6.2.1.1 Autosupervision

L'autosupervision vise à construire un espace de caractéristiques propre à un jeu de données, et ce indépendamment de la tâche que l'on cherche à accomplir sur le jeu considéré. Cet espace est généralement inféré par un encodeur. L'objectif de l'autosupervision est alors de fournir à un réseau de neurones des paramètres adaptés aux données et à partir desquels l'apprentissage d'autres tâches, comme la classification ou encore la segmentation sémantique, est rendue plus simple du fait de la réduction du nombre de paramètres à ajuster.

L'apprentissage par autosupervision d'un encodeur est réalisé par le biais d'une tâche dite « prétexte », c'est-à-dire une tâche dont la vérité terrain peut être construite de façon algorithmique, sans annotation manuelle, et dont la résolution nécessite une compréhension approfondie de la scène. De nombreuses tâches prétextes existent (re-colorisation d'une image que l'on a préalablement convertie en nuances de gris [149], détermination du positionnement relatif de deux sous-images extraites d'un même exemple [150], etc.).

La tâche prétexte que nous utilisons est celle décrite dans SimCLR [20]. On considère un encodeur E suivi d'un Perceptron multicouche g visant à projeter les représentations issues de l'encodeur, projections sur lesquelles est appliquée la fonction de coût de la tâche prétexte. L'encodeur est ici identique à celui de ResNet-18 et le Perceptron multicouche comporte deux couches. Pour chaque *batch* $B = \{x_i \mid 1 \leq i \leq N\}$ de taille N et extrait d'un jeu de données, on

construit un ensemble $\{\tilde{x}_i \mid 1 \leq i \leq 2N\}$ avec, pour tout $i \in [\![1; N]\!]$,

$$\tilde{x}_{2i-1} = t_1^{(i)}(x_i) \tag{6.1}$$

$$\tilde{x}_{2i} = t_2^{(i)}(x_i) \tag{6.2}$$

où $t_1^{(i)}$ et $t_2^{(i)}$ sont des transformations générées aléatoirement (géométrique, colorimétrique, etc.) pour chaque *batch* et qui varient à chaque époque. La génération et l'application de la transformation, pour chaque exemple, est réalisée par l'algorithme 3.

Algorithme 3 : Algorithme d'augmentation de données utilisé pour la méthode d'adaptation non supervisée et appliqué sur chaque sous-image de résolution 256×256 pixels.

Entrée : La sous-image à traiter X
$\alpha \sim \mathcal{U}(0.5, 1.5)$;
$\beta \sim \mathcal{U}(-0.15, 0.15)$;
$X \leftarrow \alpha(X - \bar{X}) + \bar{X} + \beta$ (où $\bar{X}$ est la moyenne des valeurs de X) ;

$X \leftarrow$ Miroir_horizontal(X) avec une probabilité de $\frac{1}{2}$;
$X \leftarrow$ Miroir_vertical(X) avec une probabilité de $\frac{1}{2}$;
$X \leftarrow$ Égalisation_histogramme(X) avec une probabilité de $\frac{1}{4}$;

$C \sim \mathcal{U}_D(0, 2)$;
$F \sim \mathcal{U}(0.1, 2)$;

$X \leftarrow X[:, :, C] \leftarrow 0$ avec une probabilité de $\frac{1}{4}$;
$X \leftarrow X[:, :, C] \leftarrow X[:, :, C] \times F$ avec une probabilité de $\frac{1}{4}$;
$X \leftarrow$ Ajout_bruit_Poivre_Sel(X) avec une probabilité de $\frac{1}{4}$;

$\theta \sim \mathcal{U}_D(0, 359)$;
$X \leftarrow$ Rotation(X, θ) ;

$\phi \sim \mathcal{U}(0.8, 1.2)$;
$X \leftarrow$ Redimensionnement(X, ϕ) ;
Sortie : X

Pour chaque exemple, on obtient ainsi deux exemples appariés qui représentent la même scène à une transformation près. On pose alors, pour tout $k \in [\![1; 2N]\!]$,

$$z_k = g(E(\tilde{x}_k)) \tag{6.3}$$

L'objectif est maintenant d'apprendre au réseau à rapprocher les paires dans l'espace de caractéristiques à partir des projections de l'ensemble des exemples transformés. Pour ce faire, on définit, pour tout $(i, j) \in [\![1; 2N]\!]^2$, s_{ij} comme la similarité cosinus entre z_i et z_j, c'est-à-dire que l'on pose

$$s_{ij} = \frac{z_i . z_j}{\|z_i\| . \|z_j\|} \tag{6.4}$$

On cherche alors à maximiser cette similarité au sein de chaque paire tout en la minimisant pour deux exemples issus de paires différentes. On introduit, pour tout $(i,j) \in [\![1;2N]\!]^2$, s_{ij}, la fonction de coût l définie par

$$l(i,j) = -\log \frac{\exp(s_{ij}/\tau)}{\sum_{k=1}^{2N}(1-\delta_{ik})\exp(s_{ik}/\tau)} \tag{6.5}$$

où τ est un hyper-paramètre modulant l'amplitude du coût, fixé à 10^{-1} pour nos expérimentations, et δ_{ik} est le symbole de Kronecker, valant 1 pour $i=k$ et 0 sinon. Minimiser $l(i,j)$ revient à maximiser la similarité cosinus entre z_i et z_j tout en minimisant cette même similarité entre z_i et les autres vecteurs de caractéristiques. On souhaite donc minimiser $l(2k-1,2k)$ et $l(2k,2k-1)$ pour tout $k \in [\![1;M]\!]$. La fonction de coût pour l'ensemble du réseau vaut alors

$$\mathcal{L}_{self} = \frac{1}{2M}\left(\sum_{k=1}^{M} l(2k-1,2k) + l(2k,2k-1)\right) \tag{6.6}$$

Ce procédé est réitéré jusqu'à ce que tous les exemples du jeu d'apprentissage aient été sélectionnés au sein d'un *batch*, réalisant ainsi une époque. Le principe de SimCLR est résumé en figure 6.9. Précisons que ce principe général n'est pas propre à SimCLR mais est commun à toute une famille de méthodes autosupervisées [151, 152, 153] que l'on regroupe parfois sous le terme de ***Contrastive Learning***.

6.2.1.2 Méthodologie

La méthode proposée est issue de [21] (article non publié). Elle s'articule en deux temps et est illustrée en figure 6.10.

Phase autosupervisée Dans un premier temps, on cherche à créer un espace de caractéristiques commun aux domaines source et cible. Pour ce faire, on entraîne un encodeur à l'aide de la méthode SimCLR durant 1000 époques sur un ensemble de données composé d'exemples issus des deux domaines. Pour réduire le risque de biais en faveur de l'un ou l'autre domaine, le jeu d'apprentissage comprend autant d'exemples du premier domaine que du second. À la fin de cette étape, c'est le modèle résultant de la millième itération qui sert de base pour la phase supervisée.

Phase supervisée Dans un second temps, une fois l'encodeur appris, nous remplaçons le Perceptron multicouche par un décodeur pour réaliser la segmentation sémantique des fers apparents. Ce décodeur est alors entraîné durant 1000 époques à l'aide du jeu d'apprentissage du domaine source uniquement, dont on utilise les annotations. Le critère d'optimisation du modèle est alors l'entropie croisée pondérée $\mathcal{L}_{ce}$. Pour cet apprentissage, les paramètres de l'encodeur sont alors figés afin de garantir que l'espace de caractéristiques, défini par ce même encodeur, demeure inchangé. Pour la sélection du modèle, nous

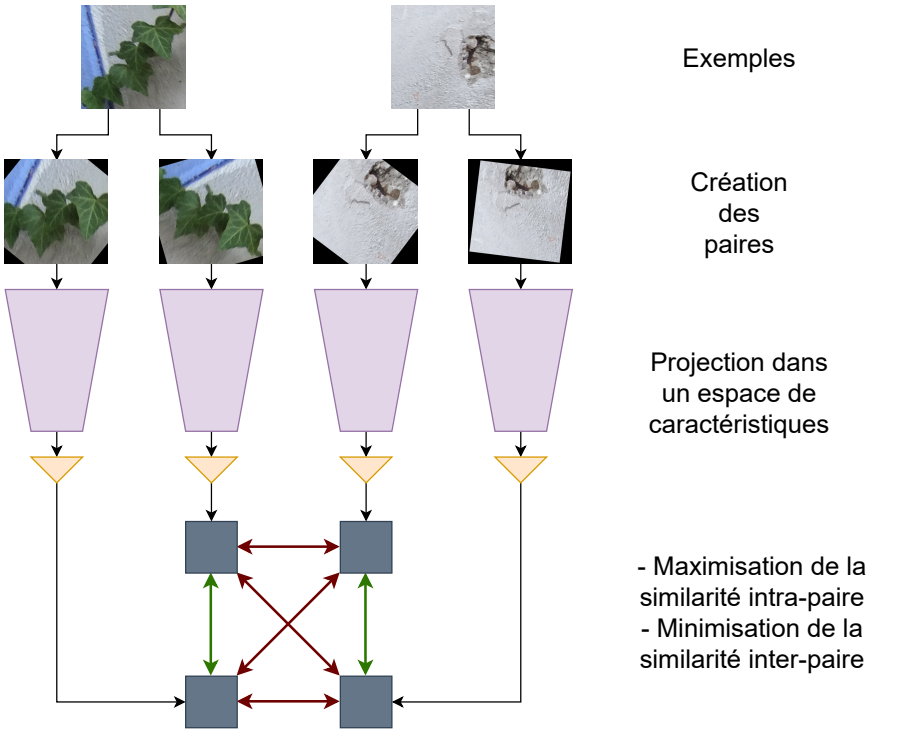


FIGURE 6.9 – Illustration de la tâche prétexte de SimCLR [20] sur un *batch* composé de deux exemples

	Apprentissage		**Validation**		**Test**	
	Piéton	Routier	Piéton	Routier	Piéton	Routier
Autosupervision	1876	1876	–	–	–	–
Supervision	1876	0	1092	0	0	418

TABLEAU 6.4 – Composition des jeux de données pour la méthode d'adaptation de domaine non supervisée. Dans la phase autosupervisée, les jeux sont constitués de sous-images de résolution 256×256 pixels et sont utilisés sans annotation. Le jeu de validation est également composé de sous-images de même dimension tandis que l'évaluation porte sur 418 images en pleine résolution.

reprenons les critères F_1 et EP utilisés jusqu'à présent. Ces derniers portent sur le jeu de validation, qui est issu du domaine source et est donc annoté.

Le tableau 6.4 résume les jeux de données utilisés pour les deux étapes de la méthode autosupervisée.

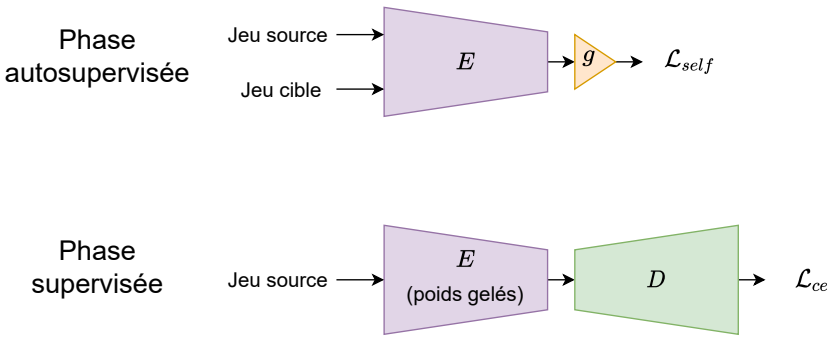


FIGURE 6.10 – Illustration des deux étapes de la méthode proposée (issue de Sun *et al.* [21])

6.2.1.3 Résultats

Les résultats quantitatifs sont présentés dans le tableau 6.5. On constate, sur le jeu de test intégral, une exactitude pondérée et un rappel plus élevés avec cette stratégie d'adaptation qu'avec une évaluation directe (voir tableau 6.3a), et ce quel que soit le critère de sélection du modèle. La précision y est cependant moindre, signifiant un plus grand nombre de fausses alarmes pour le modèle adapté. Dans cette expérimentation, le critère EP semble donner de meilleurs résultats, toutes métriques confondues. C'est, en particulier, le cas pour le rappel pixellique qui s'établit à plus de 40%. En termes de rappel par composantes connexes 6.11, on observe qu'on détecte largement plus de fers apparents avec le critère EP qu'avec le critère F_1, et ce pour tous les seuils d'admission. Ainsi, plus de la moitié des fers apparents sont détectés avec un seuil d'admission de 50% pour le critère EP tandis qu'on en relève moins de 5% pour ce même seuil avec le critère F_1.

Qualitativement parlant, on relève, en cohérence avec la faible précision mesurée, de nombreux faux positifs (voir figure 6.12). Si les fers apparents sont partiellement détectés, le réseau semble considérer les revêtements à la texture granuleuse comme fers apparents. Ce point est particulièrement visible sur le joint du dernier exemple. Une telle erreur pourrait s'expliquer par le fait que le joint, rectiligne, est entouré d'un revêtement dont l'aspect se rapproche d'une perte de matière. L'ensemble peut ainsi rappeler la structure d'un fer apparent.

Dans la visualisation des représentations (*cf.* figure 6.13), on constate que les deux domaines sont parfaitement entremêlés en sortie de l'encodeur, là où ils étaient nettement distincts pour l'encodeur issu du modèle appris sur le tunnel piéton. On peut également souligner que les anomalies, particulièrement groupées au sein de chaque domaine avant adaptation, sont davantage dispersées. La méthode d'adaptation a donc réussi à créer un espace de caractéristiques commun aux deux domaines. Néanmoins, les résultats, aussi bien quantitatifs que qualitatifs, demeurent de faible qualité, même s'ils sont globalement meilleurs que ceux du modèle n'ayant pas été adapté. La visualisation t-SNE

des représentations nous laisse ainsi conjecturer que l'autosupervision a engendré des « erreurs d'appariement » au sein de cet espace de caractéristiques, des revêtements sains d'un des domaines se trouvant « proches » d'anomalies de l'autre domaine.

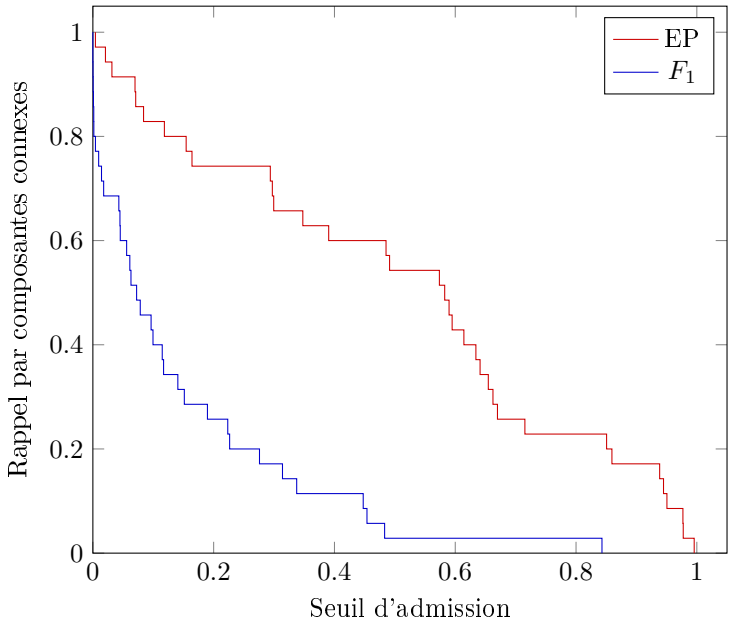
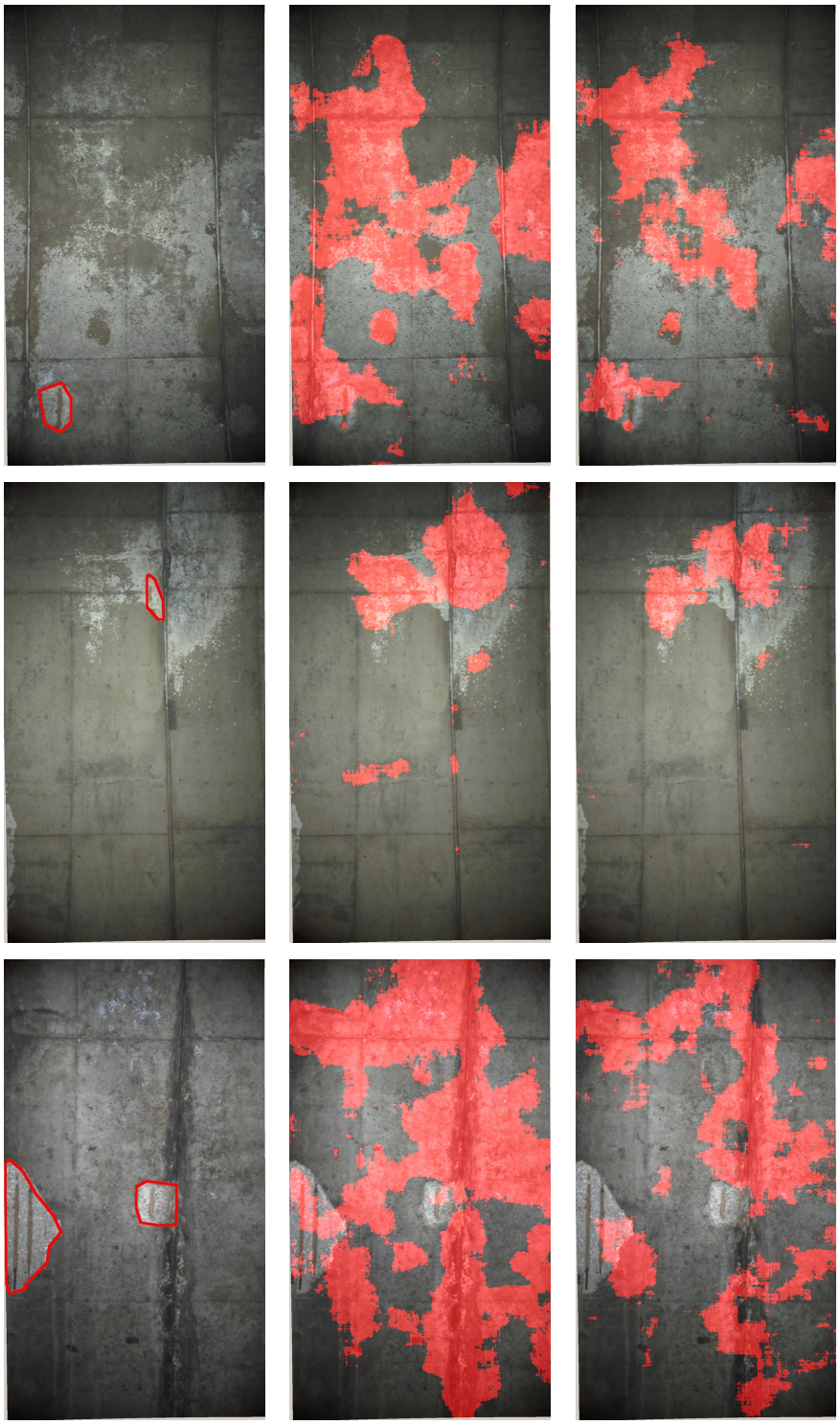


FIGURE 6.11 – Rappel par composantes connexes pour le modèle appris par la méthode non supervisée et pour les deux critères de sélection de modèle.

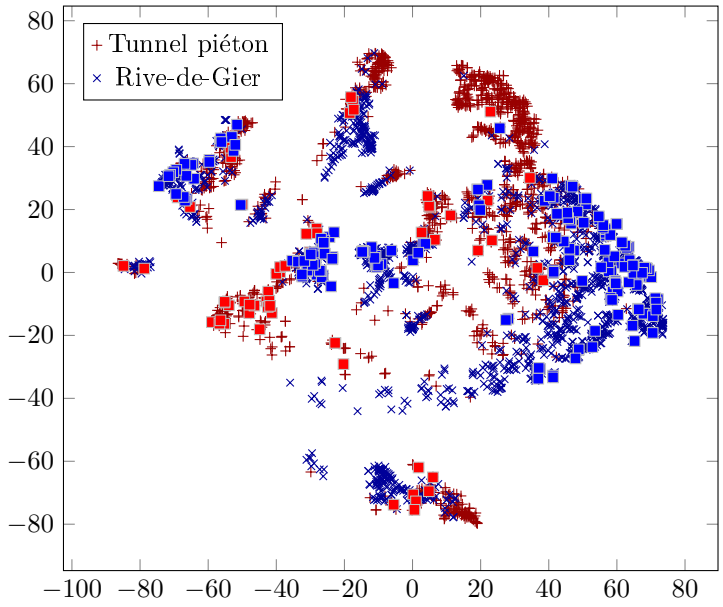
	EP	P	R	F_1
Modèle non adapté, critère EP	52.65	**00.42**	06.81	**00.78**
Modèle non adapté, critère F_1	50.18	00.20	00.68	00.30
Adaptation non supervisée, critère EP	**62.38**	00.22	**42.51**	00.44
Adaptation non supervisée, critère F_1	53.97	00.19	15.80	00.37
Modèle non adapté, critère EP (R)	52.63	**06.11**	06.81	**06.44**
Modèle non adapté, critère F_1 (R)	50.19	03.29	00.68	01.13
Adaptation non supervisée, critère EP (R)	**57.24**	02.21	**42.51**	04.19
Adaptation non supervisée, critère F_1 (R)	50.06	01.48	15.80	02.70

TABLEAU 6.5 – Métriques obtenues pour la méthode non supervisée (EP : exactitude pondérée ; P : précision ; R : rappel ; F_1 : F_1-score) – (R) : évaluation sur le jeu de test restreint de Rive-de-Gier. Les résultats du modèle non adapté sont repris du tableau 6.3a

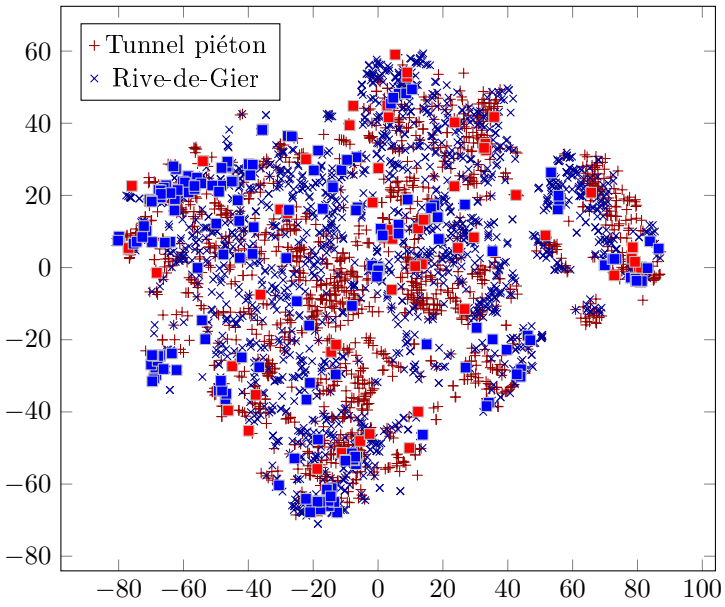


Vérité terrain | Critère EP | Critère F_1

FIGURE 6.12 – Résultats obtenus pour la méthode non supervisée selon les critères EP et F_1.



(a) Apprentissage supervisé



(b) Méthode non supervisée

FIGURE 6.13 – Comparaison des représentations issues de l'apprentissage supervisé sans adaptation et de la méthode non supervisée introduite par Sun *et al.* [21] sur les exemples de la figure 6.8. Les carrés pleins représentent les voisinages 32×32 comprenant un fer apparent.

6.2.2 Adaptation de domaine faiblement supervisée

Résumé du protocole expérimental

- **Architecture** SegNet
- **Poids initiaux** Variables (voir texte)
- **Époques** 1000 • **Optimiseur** *Adam*
- **Augmentation de données** Drouyer [130]
- **Critère d'optimisation** Entropie croisée pondérée
- **Critère de sélection des modèles à évaluer**
  - Modèle issu de la dernière époque
- **Objectifs**
  - Évaluer une approche faiblement supervisée d'adaptation de domaine, requérant d'annoter (au moins) un exemple du domaine cible
  - Évaluer l'influence du choix de l'exemple du domaine cible utilisé pour guider le processus d'adaptation

Après avoir évalué une stratégie d'adaptation de domaine non supervisée, nous expérimentons une stratégie d'adaptation de domaine faiblement supervisée. Plus précisément, il s'agit d'une méthode pour laquelle il est nécessaire d'annoter au moins une image du domaine cible, ceci afin de « calibrer » le modèle et, ainsi, l'adapter au tunnel constituant le domaine cible. Pour réaliser cette opération, nous agissons au niveau des couches de *batch normalization* (BN), introduites dans le chapitre 3. Pour rappel (*cf.* chapitre 3, section 3.1.1.5), une telle couche est définie comme une application

$$\begin{aligned} BN : \mathcal{I}_{W,H,D} &\rightarrow \mathcal{I}_{W,H,D} \\ f &\mapsto \gamma \frac{f - \hat{\mu}}{\sqrt{\hat{\sigma} + \varepsilon}} + \beta \end{aligned} \tag{6.7}$$

où γ et β sont appris lors de la phase de rétropropagation (*backward*) et $\hat{\sigma}$ ainsi que $\hat{\mu}$ sont estimés lors de la phase de propagation (*forward*).

6.2.2.1 Adaptation de domaine et normalisation

L'idée de mettre à profit les couches de *batch normalization* pour l'adaptation de domaine n'est pas nouvelle. S'il n'y a pas d'hypothèse unique pour expliquer l'intérêt que porte les auteurs aux couches de cette nature pour l'adaptation de domaine, on peut toutefois souligner que ces dernières ne représente qu'une proportion très limitée des paramètres du modèle (généralement moins de 5%) et sont généralement considérées comme indépendantes des couches de convolution, dans le sens où elle ne font qu'appliquer une transformation affine (la même pour chaque pixel) sur les caractéristiques entrantes normalisées.

En 2017, une première méthode a été introduite par Li *et al.* [154]. Elle consiste à ajuster, de manière non supervisée, les estimations de $\hat{\mu}$ et $\hat{\sigma}$ de chacune des couches de BN pour qu'elles correspondent aux distributions de

probabilité du domaine cible au sein des espaces de caractéristiques transformés par ces couches. Les autres paramètres du réseau demeurent alors inchangés, y compris ceux des couches de normalisation appris par rétropropagation, à savoir γ et β. Si le nombre de paramètres concernés par cette méthode est faible, elle permet cependant de calibrer rapidement un modèle déjà appris sur un domaine.

Plus récemment, Chang *et al.* [155] ont proposé la méthode $DSBN$ (*Domain-specific batch normalization*), dans laquelle les couches de normalisation adoptent un comportement différent selon le domaine dont proviennent les données. Une telle couche N est alors définie comme un couple (BN_1, BN_2) tel que

$$N(x) = \begin{cases} BN_1(x) & \text{si } x \text{ provient du domaine source} \\ BN_2(x) & \text{si } x \text{ provient du domaine cible} \end{cases} \tag{6.8}$$

Ainsi, on cherche à construire un opérateur de normalisation propre à chaque domaine, tout en conservant les caractéristiques extraites par les couches de convolution. Cette méthode permet d'agir sur l'ensemble des paramètres des couches de BN mais induit, en revanche, un apprentissage plus contraignant. En effet, ce dernier porte simultanément sur les jeux source et cible, ce qui en rallonge la durée et requiert la disponibilité des données annotées du domaine source au moment de l'adaptation. De plus, comme cet apprentissage est supervisé, et que le domaine cible est dépourvu d'annotations, il faut disposer d'un modèle initial meilleur que le hasard sur ce même domaine. Ce modèle, appelé « pseudo-annotateur », est utilisé pour générer les prédictions pour les exemples de ce jeu, qui sont alors employées comme vérités terrain. Au fur et à mesure de l'apprentissage, l'importance accordée à la fonction de coût concernant ces annotations, imparfaites, diminue.

Indépendamment du contexte de l'adaptation de domaine, certains travaux ont mis en évidence l'expressivité qui caractérise les couches de BN, alors même que ces dernières ne couvrent qu'une faible proportion du nombre de paramètres d'un réseau de neurones. Dans [156], Frankle *et al.* réalisent l'expérience suivante : considérant un réseau convolutif dont les poids sont initialisés aléatoirement, ils opèrent un apprentissage supervisé, tout en restreignant les paramètres appris à ceux des couches de BN, figeant ainsi la valeur des autres avant même le début de l'apprentissage. Ces derniers conservent donc la valeur aléatoire qui leur a été attribuée. Après deux expérimentations menées séparément sur ImageNet et sur CIFAR-10, ils montrent que les modèles appris de cette manière présentent des performances remarquablement élevés compte tenu du faible nombre de paramètres que représentent les couches de BN. Ces résultats demeurent cependant inférieurs à ceux des modèles dont l'apprentissage a porté sur l'intégralité des paramètres (dans leurs travaux, les auteurs réussissent à atteindre une exactitude pondérée située 10 points (%) en-dessous de celle réalisant l'état de l'art sur CIFAR-10).

Sur la base de ces travaux, nous cherchons à construire une méthode d'adaptation de domaine qui allie la praticité de [154] (calibration via un apprentissage sur le domaine cible uniquement) avec l'efficacité de [155] (ajustement de l'in-

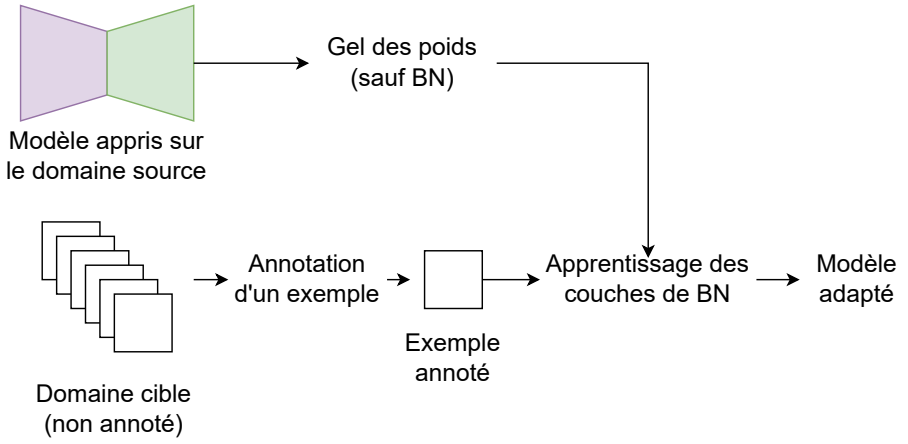


FIGURE 6.14 – Stratégie d'adaptation de domaine faiblement supervisée mise en œuvre.

tégralité des paramètres des BN, dont Frankle *et al.* [156] ont montré qu'il permettait aux BN d'avoir une influence considérable sur le comportement du modèle). Pour ce faire, nous implémentons et évaluons une approche faiblement supervisée, requérant l'annotation d'au moins une image du domaine cible par un opérateur.

6.2.2.2 Méthodologie

La méthode d'adaptation de domaine proposée est la suivante : sur la base d'un nombre réduit d'images annotées du domaine cible et d'un modèle appris sur le domaine source, nous réalisons l'apprentissage d'une architecture de segmentation sur 1000 époques ne modifiant que les paramètres des couches de *batch normalization* (*i.e.* $\hat{\mu}$, $\hat{\sigma}$, γ et β). Pour cette expérimentation, nous avons considéré le cas particulier où une seule image est annotée. Cette stratégie est illustrée en figure 6.14.

L'utilisation d'une quantité plus importante d'images annotées permettrait assurément d'atteindre de meilleurs scores sur le jeu de test. Néanmoins, il convient de noter que, dans ce protocole expérimental, toutes les images annotées utilisées doivent nécessairement être issues de ce même jeu de test. Cela correspond, en effet, à la situation rencontrée d'un point de vue opérationnel dans laquelle les jeux cibles représentent les tunnels nouvellement acquis. Il est donc important de limiter le nombre d'images considérées afin de réduire l'intersection entre les jeux d'apprentissage et, ainsi, que l'évaluation du modèle ne s'en trouve trop favorablement biaisée. Symétriquement, il aurait été possible d'évaluer le modèle sur le jeu de test auquel les images servant à l'adaptation auraient été retranchées, mais cela nous aurait contraint à réévaluer l'ensemble des modèles entraînés jusqu'à présent sur ce même jeu réduit pour pouvoir en dresser une comparaison.

FIGURE 6.15 – Image, issue du tunnel de Rive-de-Gier, utilisée avec annotation pour l'adaptation des modèles sur ce tunnel. Les fers apparents sont détourés en rouge.

Comme nous ne disposons pas de jeu de validation sur le domaine cible, c'est le millième modèle qui est considéré. L'image sélectionnée est découpée en 28 sous-images de taille 256×256 pixels et nous utilisons l'entropie croisée pondérée comme fonction de coût. Les coefficients de cette fonction sont déterminés sur la base de l'image annotée du domaine cible. La composition des jeux de données est présentée dans le tableau 6.6.

Dans le cadre de cette méthode, il est important que l'image du domaine cible à annoter soit choisie avec soin. En effet, puisque cette image est le seul exemple de ce jeu que verra le modèle lors de son adaptation, il faut qu'elle soit la plus représentative possible du domaine cible. De plus, il est impératif que l'anomalie recherchée y figure. L'image de Rive-de-Gier retenue pour ce rôle est présentée en figure 6.15.

Pour les modèles servant à l'initialisation de la méthode, nous reprenons ceux utilisés lors de l'évaluation de l'influence du biais inter-domaine. Pour celui appris sur le tunnel piéton, nous prenons pour base celui maximisant

Apprentissage	Validation	Test
28	–	418

TABLEAU 6.6 – Composition des jeux de données pour la méthode d'adaptation de domaine faiblement supervisée. Le jeu d'apprentissage correspond à l'image du domaine cible découpée en 28 sous-images de résolution 256×256 pixels. Comme pour la méthode non supervisée, l'évaluation porte sur 418 images en pleine résolution.

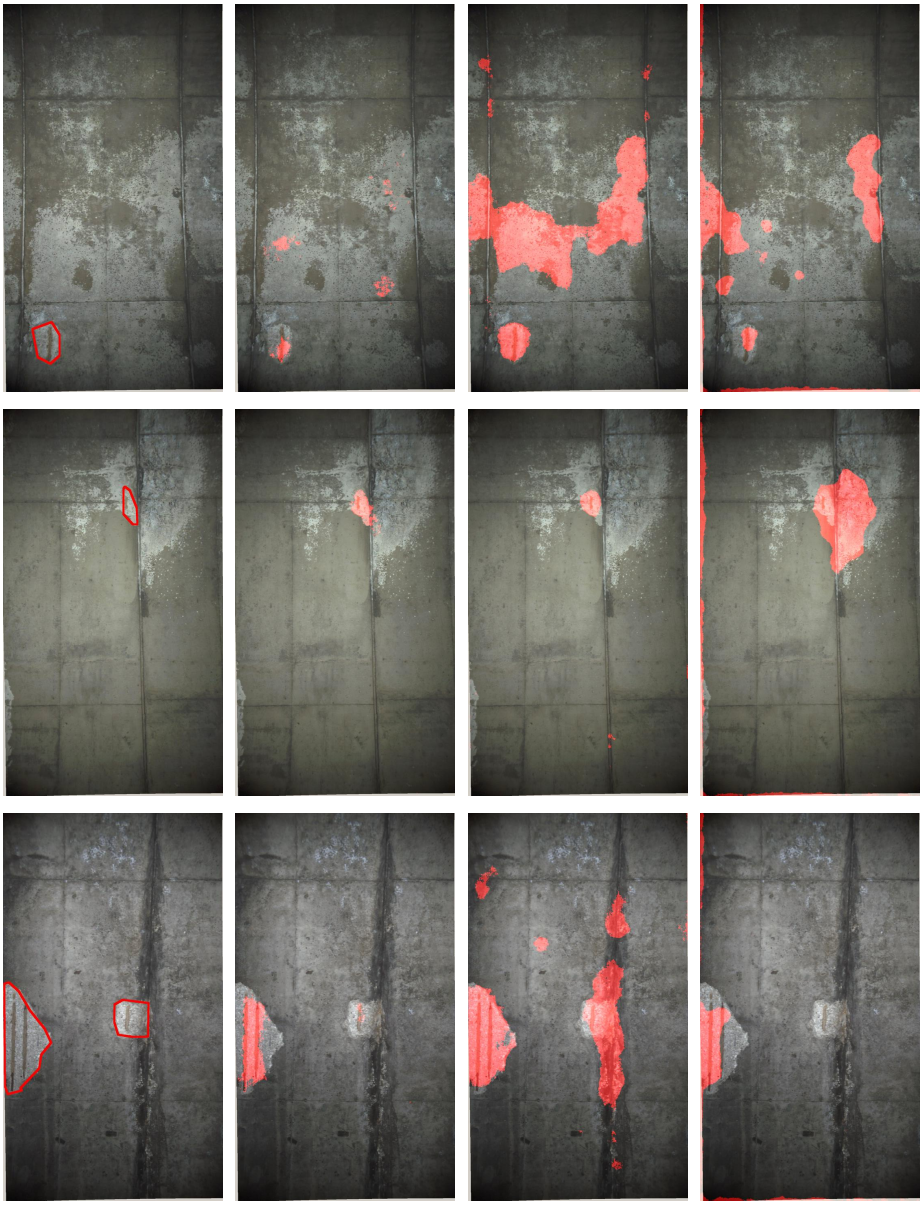
l'exactitude pondérée sur le jeu de validation. Ce modèle présente, comme nous l'avons vu précédemment, des performances supérieures au modèle maximisant le F_1-score sur ce même jeu de validation, toutes métriques confondues (voir tableau 6.3a). Quant au modèle appris sur la base CODEBRIM, les critères EP et F_1 désignent un même modèle. C'est donc ce dernier qui est utilisé.

Le protocole d'adaptation étant construit sur la base de l'expérience réalisée par Frankle *et al.* [156], nous évaluons également la configuration employée par ses auteurs. Ainsi, en plus des modèles appris sur le tunnel piéton et sur la base CODEBRIM, nous considérons un modèle dont les poids sont initialisés au hasard et immédiatement figés, à l'exception des paramètres des BN. Cette comparaison nous permet de mesurer l'apport éventuel d'un modèle spécialisé dans la tâche d'intérêt pour l'adaptation de domaine.

6.2.2.3 Résultats

Les résultats quantitatifs sont présentés dans le tableau 6.7 et trois exemples de résultats sont visualisés en figure 6.16. Il en ressort que les trois configurations se positionnent au-dessus de l'ensemble des modèles appris par supervision sur d'autres sites et testés sur Rive-de-Gier, et ce quelle que soit la métrique considérée (voir tableau 6.3). Aussi, ces résultats mettent en avant l'intérêt d'un modèle dédié à la segmentation des fers apparents comme point de départ du processus d'adaptation. En effet, le modèle initialisé aléatoirement est le seul à considérer, à tort, les artefacts sur les bords des images comme fers apparents, erreur qui est répétée pour chaque exemple. Là où le modèle pré-appris sur la base CODEBRIM tend à sur-détecter les anomalies quand celui repris du tunnel piéton les sous-détecte, le modèle initialisé aléatoirement présente les défauts de ces deux modèles. L'image présentée en première ligne de la figure 6.16 illustre ce phénomène : le fer apparent n'est que partiellement reconnu, comme avec le modèle issu du tunnel piéton, tandis que la quantité de fausses alarmes se rapproche du modèle appris sur la base CODEBRIM. Le rappel par composantes connexes (figure 6.17) de ce dernier modèle est, en effet, supérieur à celui des deux autres, et ce pour tous les seuils d'admission inférieurs à 80%. Pour les seuils d'admission supérieurs à 80%, c'est le modèle initialisé aléatoirement qui présente le rappel par composantes connexes le plus élevé parmi les trois modèles. Cela traduit le fait qu'un petit nombre de fers apparents, comme celui présents sur l'image de la seconde ligne de la figure 6.16, font l'objet d'une sur-détection telle que la zone prédite comme fer apparent inclus intégralement l'anomalie.

Les représentations t-SNE des caractéristiques induites pour chaque modèle sont présentées en figure 6.18. On peut noter que, pour les configurations se basant sur les modèles appris sur le tunnel piéton (figures 6.18b et 6.18c) et sur la base CODEBRIM (figures 6.18d et 6.18e), les distributions des domaines source et cible concordent davantage avec cette stratégie d'adaptation que sur les modèles initiaux. On en déduit alors que les caractéristiques extraites par les couches de convolution (et dont les paramètres n'ont pas été altérés par la méthode d'adaptation de domaine), demeurent plus efficaces pour l'obtention



Vérité terrain Tunnel piéton CODEBRIM Modèle aléatoire

FIGURE 6.16 – Résultats obtenus pour la méthode faiblement supervisé selon le modèle de base choisi. Les fers apparents sont détourés en rouge dans la vérité terrain.

	EP	P	R	F_1
Modèle non adapté, critère EP	52.65	00.42	06.81	00.78
Modèle non adapté, critère F_1	50.18	00.20	00.68	00.30
Adaptation supervisée, hasard	81.59	01.87	66.41	03.64
Adaptation supervisée, tunnel piéton	80.85	**17.40**	61.97	**27.17**
Adaptation supervisée, CODEBRIM	**87.84**	02.11	**79.10**	04.11
Modèle non adapté, critère EP (R)	52.63	06.11	06.81	06.44
Modèle non adapté, critère F_1 (R)	50.19	03.29	00.68	01.13
Adaptation supervisée, hasard (R)	80.79	16.98	66.41	27.05
Adaptation supervisée, tunnel piéton (R)	80.65	**57.91**	61.97	**59.87**
Adaptation supervisée, CODEBRIM (R)	**87.40**	21.43	**79.10**	33.73

TABLEAU 6.7 – Synthèse des métriques obtenues pour les différentes configurations de la méthode d'adaptation faiblement supervisée. (EP : exactitude pondérée ; P : précision ; R : rappel ; F_1 : F_1-score) – (R) : évaluation sur le jeu de test restreint de Rive-de-Gier. Les résultats du modèle non adapté sont repris du tableau 6.3a

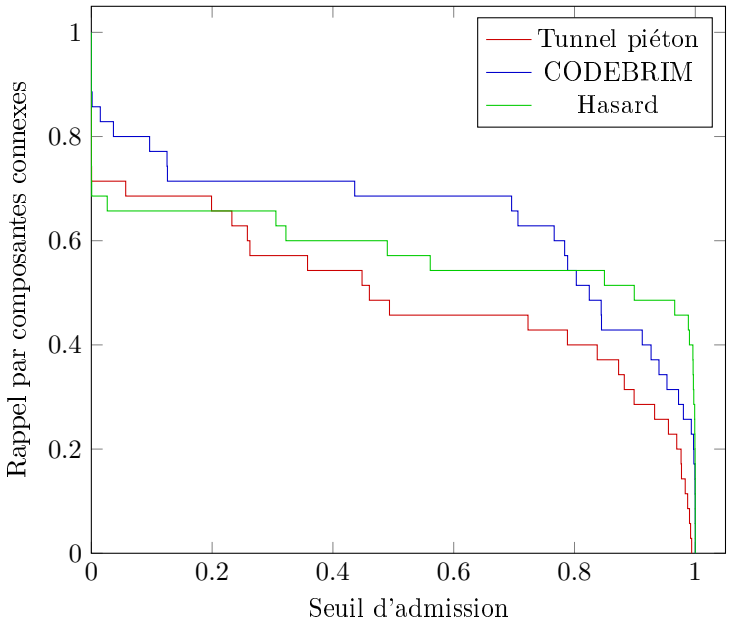
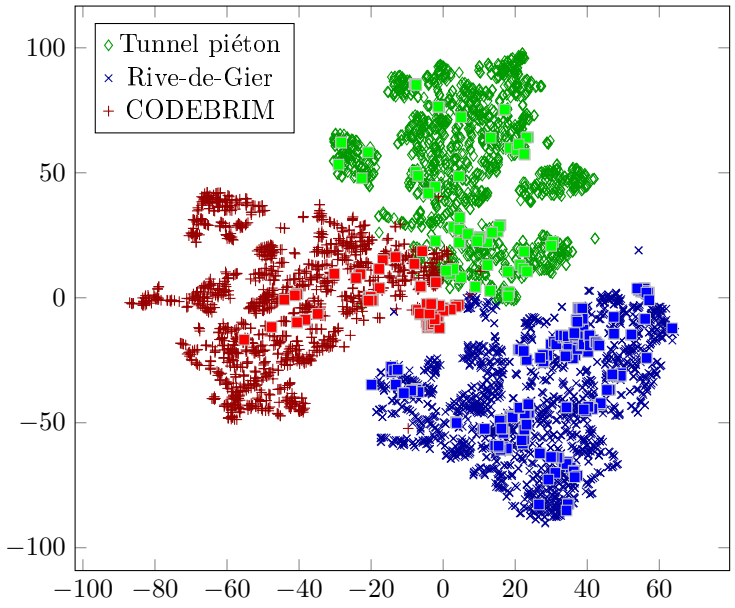


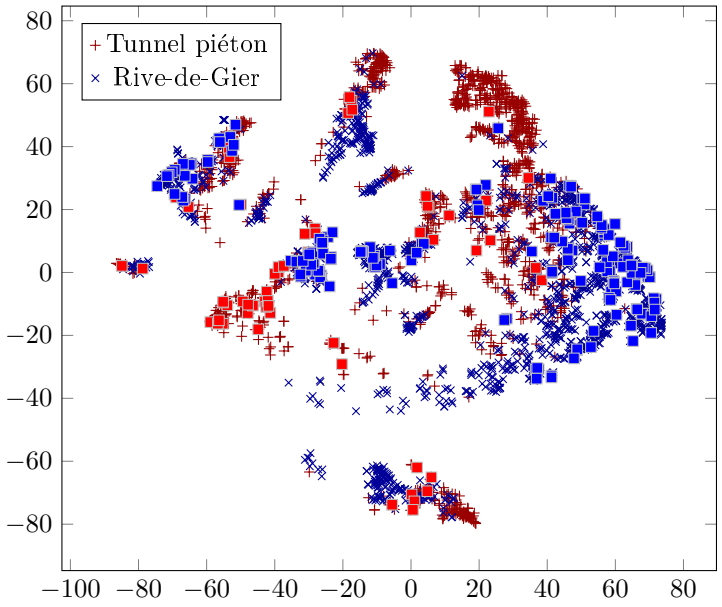
FIGURE 6.17 – Rappel par composantes connexes pour le modèle appris par la méthode faiblement supervisée et selon les trois modèles sources.

de modèles multi-sites que celles produites par des convolution aléatoires (figure 6.18a). En effet, pour cette dernière configuration, on peut voir que les trois domaines sont bien différenciés, presque linéairement séparables deux à deux.

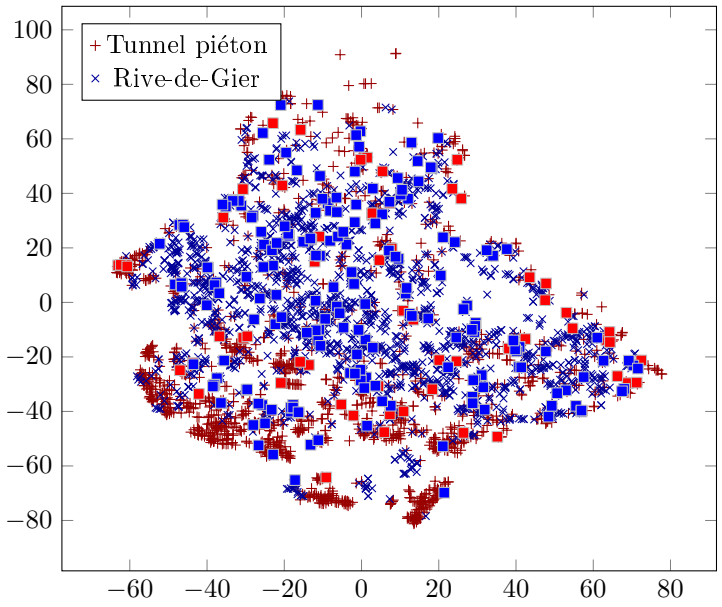


(a) Adaptation BN, hasard

FIGURE 6.18 – Comparaison des représentations issues de l'apprentissage supervisé sans adaptation et de la méthode d'adaptation faiblement basée sur les couches de *batch normalization*. Les carrés pleins représentent les voisinages 32×32 comprenant un fer apparent.

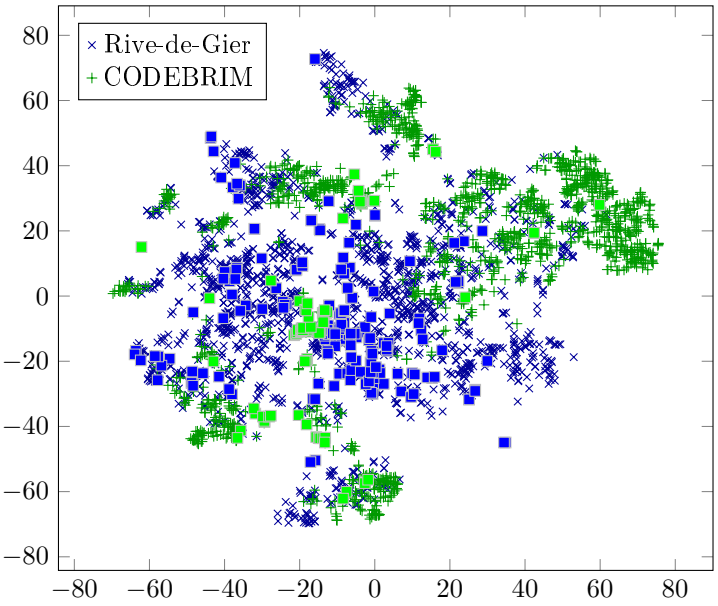


(b) Apprentissage supervisé, tunnel piéton

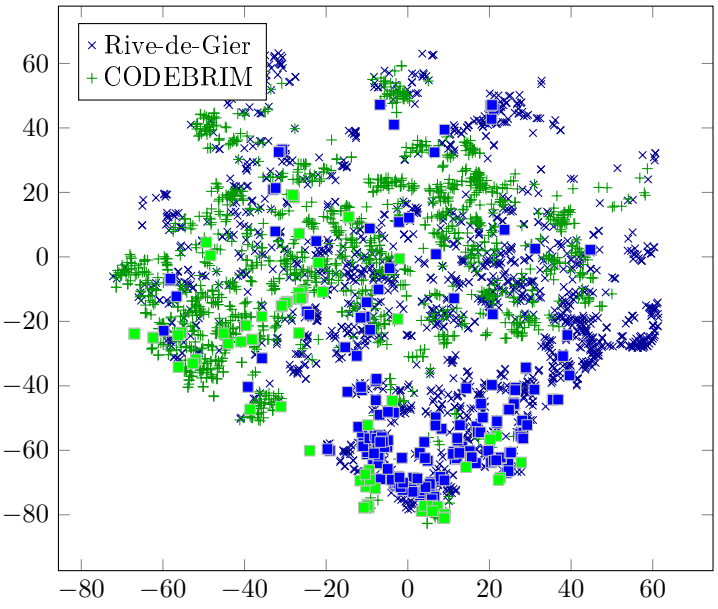


(c) Adaptation BN, tunnel piéton

FIGURE 6.18 – Comparaison des représentations issues de l'apprentissage supervisé sans adaptation et de la méthode d'adaptation faiblement basée sur les couches de *batch normalization*. Les carrés pleins représentent les voisinages 32×332 comprenant un fer apparent.



(d) Apprentissage supervisé, CODEBRIM



(e) Adaptation BN, CODEBRIM

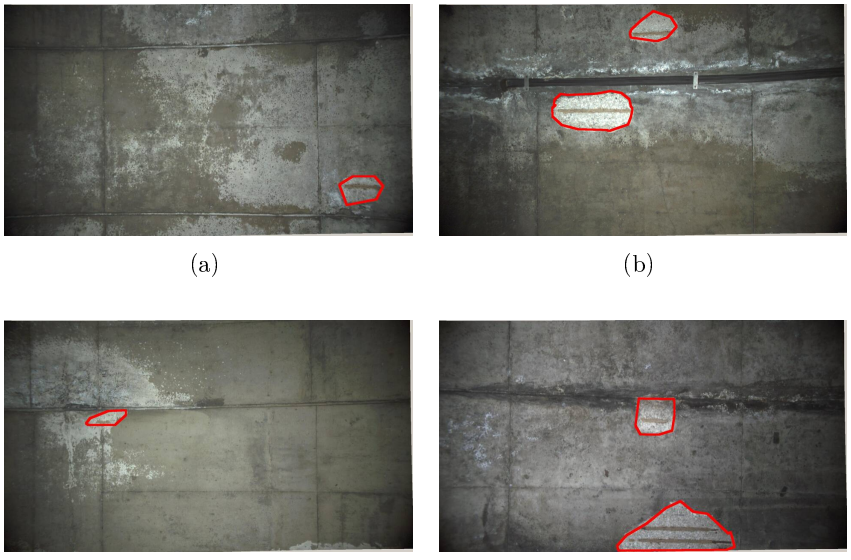
FIGURE 6.18 – Comparaison des représentations issues de l'apprentissage supervisé sans adaptation et de la méthode d'adaptation faiblement basée sur les couches de *batch normalization*. Les carrés pleins représentent les voisinages 32×32 comprenant un fer apparent.

6.2.2.4 Influence du choix de l'image

Pour mesurer l'influence du choix de l'image sur les performances du modèle adapté, nous répétons le processus d'adaptation faiblement supervisé en faisant varier l'image servant à l'adaptation. Nous considérons quatre exemples : les trois exemples de test de Rive-de-Gier utilisés dans ce chapitre ainsi que l'exemple choisi pour réaliser l'adaptation faiblement supervisée. La figure 6.19 reprend ces différents exemples.

Concernant les modèles initiaux sur lesquels porte l'adaptation, nous choisissons d'adapter des modèles dont les poids sont tirés aléatoirement. Ces poids sont réinitialisés avant chaque apprentissage, si bien que tous les modèles de départ considérés dans cette expérimentation sont différents. Reprendre un modèle appris sur un autre domaine comme point de départ de l'algorithme permettrait probablement d'obtenir de meilleurs résultats. Néanmoins, ce choix introduirait un biais dans l'étude. En effet, on mesurerait alors les performances de notre méthode par rapport à une instance de modèle en particulier. La portée des conclusions qui en découleraient s'en verrait alors réduite.

Également, pour renforcer la significativité des résultats, nous répétons l'ensemble du procédé d'adaptation cinq fois de sorte à obtenir, pour chaque image, un échantillon de cinq modèles adaptés.



(a) (b)

(c) (d)

FIGURE 6.19 – Images considérées dans l'étude de l'influence du choix de l'exemple sur les performances des modèles adaptés. Les fers apparents sont détourés en rouge. L'image (b) correspond à l'image utilisée dans la section 6.2.2.2.

À partir de chaque échantillon, nous avons calculé la moyenne et l'écart-

type pour les quatre métriques pixelliques considérées (exactitude pondérée, précision, rappel et F_1-score). Les résultats sont reportés dans le tableau 6.8. On peut y relever une grande disparité entre échantillons :

Image (a) Avec une précision moyenne inférieure à 1% sur le jeu intégral (10% sur le jeu restreint) et un écart-type inférieur à 0,1 (écart-type inférieur à 1 sur le jeu restreint), il ressort que les modèles appris avec cette image génèrent systématiquement un grand nombre de fausses alarmes au regard du nombre de fers apparents présents. Certains modèles de l'échantillon parviennent cependant à détecter la majeure partie de ces anomalies, puisque le rappel moyen s'établit à près de 60%.

Image (b) L'image (b) permet aux modèles d'atteindre une exactitude pondérée moyenne de 80% environ sur les deux jeux testés avec, à chaque fois, un écart-type inférieur à 1. Concernant la précision moyenne, elle s'établit à 2,33% sur le jeu intégral. Sur le jeu restreint, ce sont les modèles adaptés à partir de l'image (b) qui affichent la meilleure précision moyenne (plus de 20%) et qui génèrent ainsi le moins de fausses alarmes parmi les configurations testées. Quant au rappel moyen, il se situe à près de 63% et présente un écart-type de 2,36.

Image (c) Les modèles adaptés à partir de l'image (c) présentent de faibles performances. En termes d'exactitude pondérée, cette configuration présente une exactitude pondérée moyenne proche de 50% avec un écart-type de l'ordre de 0,5. Quant à la précision et au rappel moyens, ils se positionnent systématiquement en-dessous de 5%, avec là encore un écart-type inférieur à 1. On peut ainsi en déduire que chaque modèle de l'échantillon a un comportement proche de celui d'un modèle répondant au hasard.

Image (d) Parmi les 4 images testées, c'est l'image (d) qui permet aux modèles adaptés d'obtenir la meilleure exactitude pondérée ainsi que le meilleur rappel avec, en outre, un écart-type inférieur aux images (a) et (b) pour cette dernière métrique. Il s'ensuit que cet échantillon contient les modèles qui détectent le plus de fers apparents.

En comparant ces différentes métriques avec le nombre de pixels représentant des fers apparents dans l'image associée (*cf.* figure 6.20), on observe que les métriques moyennes semblent positivement corrélées avec la proportion de fers apparents visibles dans l'image. Dans le détail, l'exactitude pondérée et le rappel moyens augmentent continuellement lorsque la proportion de l'image représentant des fers apparents augmente. Pour la précision et le F_1-score moyen, cette croissance est notable pour les trois premières images mais ces deux métriques baissent légèrement pour l'image (d). Par ailleurs, les images (b) et (d), qui sont celles qui permettent de réaliser les meilleures performances moyennes, sont les seules à présenter deux fers apparents avec, à chaque fois, l'un d'eux dans la partie centrale de l'image, légèrement surexposée, et l'autre sur l'un des bords de cette dernière.

Même s'il est difficile de tirer des conclusions définitives sur la base de cinq apprentissages par image, il ressort de cette expérimentation que les images

Exemple annoté	EP		P		R		F_1	
	μ	σ	μ	σ	μ	σ	μ	σ
Image (a)	74.75	3.76	0.60	0.07	58.62	8.11	1.18	0.14
Image (b)	80.17	0.92	2.33	0.47	62.93	2.36	4.49	0.87
Image (c)	50.10	0.51	0.13	0.06	01.95	0.22	0.24	0.10
Image (d)	84.25	0.88	2.92	1.34	71.24	1.86	5.56	2.46

(a) Jeu intégral

Exemple annoté	EP		P		R		F_1	
	μ	σ	μ	σ	μ	σ	μ	σ
Image (a)	74.60	3.56	08.47	0.51	58.62	8.11	14.77	0.92
Image (b)	79.38	0.35	20.73	7.05	62.93	2.36	30.37	7.48
Image (c)	50.28	0.30	02.32	0.80	01.95	0.22	02.06	0.43
Image (d)	82.27	1.43	16.44	6.85	71.24	1.86	26.00	8.96

(b) Jeu restreint

TABLEAU 6.8 – Résultats de notre méthode d'adaptation faiblement supervisée en fonction de l'image initiale choisie. Pour chaque métrique, on calcule la moyenne (μ) et l'écart-type (σ) sur l'ensemble de l'échantillon associé. (EP : exactitude pondérée ; P : précision ; R : rappel ; F_1 : F_1-score)

qui contiennent beaucoup de fers apparents (que ce soit en nombre de pixels ou en nombre de composantes connexes) sont celles qui, en moyenne, tendent à donner de meilleurs résultats.

6.3 Synthèse

La figure 6.21 présente la synthèse des expérimentations menées sur le tunnel de Rive-de-Gier (comme domaine cible) dans le cadre de l'adaptation de domaine. En plus d'y reporter les résultats des modèles appris sur un autre domaine sans avoir bénéficié d'adaptation, nous indiquons également les résultats obtenus pour le modèle appris sur Rive-de-Gier, afin d'y servir de point de comparaison. Ce dernier ayant été entraîné sur 1010 images issues de ce même tunnel (*cf.* composition des jeux de données en tableau 5.3 ou en tableau 2.2), il fournit un résultat de référence et permet ainsi le positionnement relatif des autres approches. Nous qualifions de « métrique de référence » toute métrique établie pour ce modèle sur le tunnel de Rive-de-Gier.

Sur le jeu intégral, on peut relever que les modèles obtenus par adaptation faiblement supervisée admettent une exactitude pondérée et un rappel proches des valeurs de référence. Sur les trois modèles adaptés par la technique d'ajustement des paramètres des BN à partir de l'image présentée en figure 6.15, celui utilisant le modèle appris sur la base CODEBRIM pour l'initialisation des poids parvient même à dépasser cette valeur. Sur la précision, en revanche, ils demeurent tout trois largement inférieurs à la précision de référence. Seul

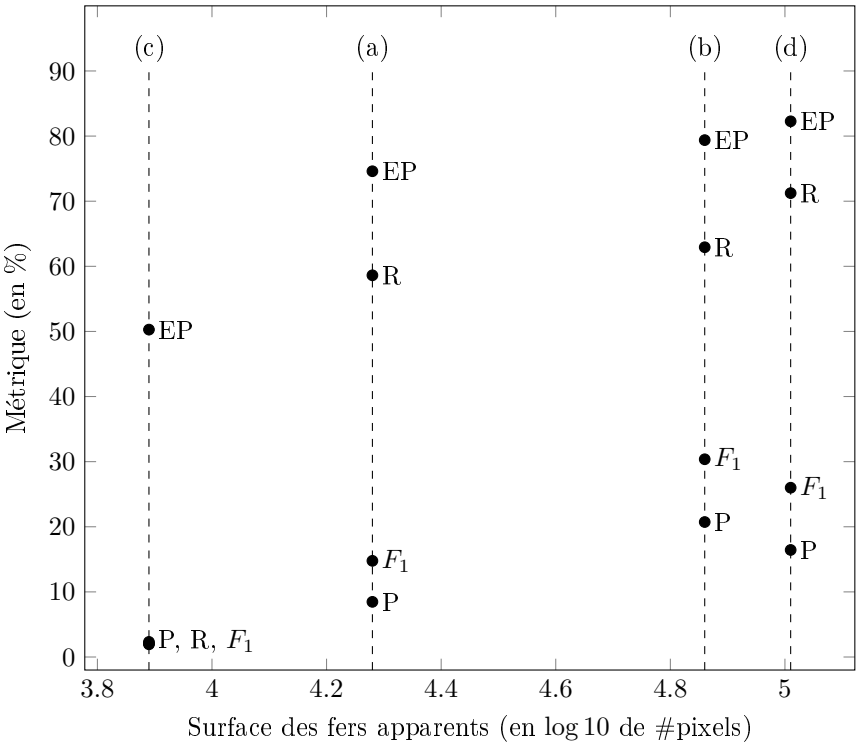


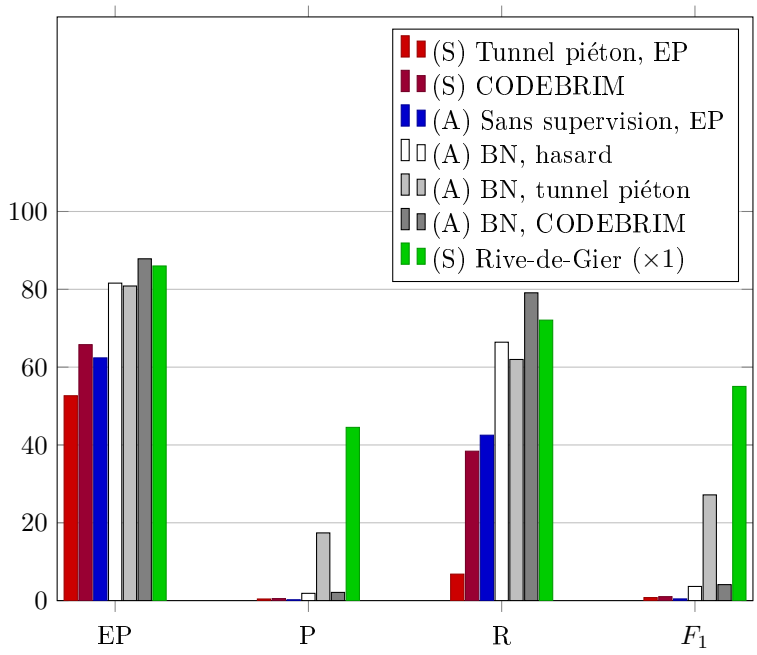
FIGURE 6.20 – Comparaison entre les résultats obtenus pour chaque image et la surface occupée par les fers apparents au sein de ces dernières. Pour l'image (c), la précision, le rappel et le F_1-score de l'échantillon associé ne sont pas différenciables dans cette représentation.

le modèle issu du tunnel piéton admet une précision environ égale à un tiers de cette dernière et un F_1-score se positionnant à plus de 50% du F_1-score de référence. Toutes les autres configurations sont nettement en dessous, y compris celle correspondant au modèle adapté sans supervision, et ce pour toutes les métriques.

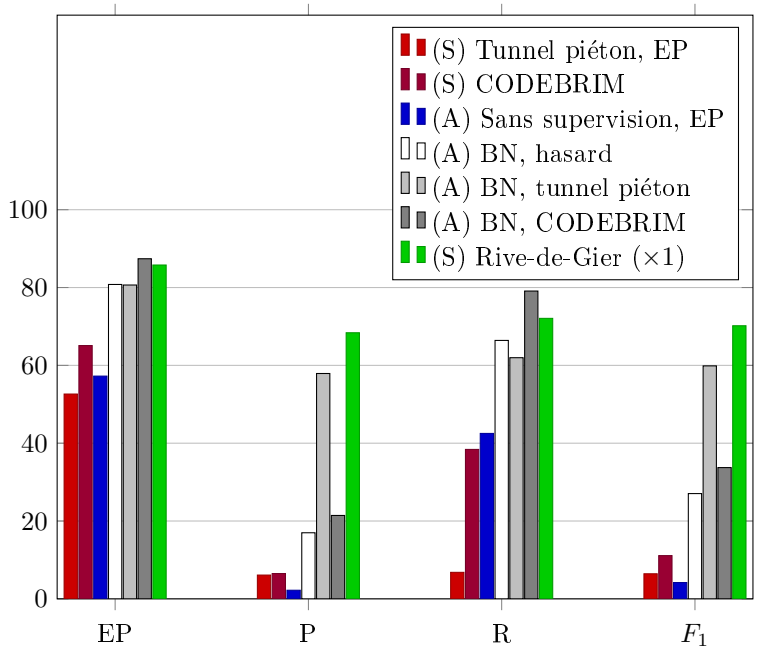
Sur le jeu restreint, on note peu d'évolution pour l'exactitude pondérée des modèles par rapport au jeu intégral. Pour la précision, en revanche, les résultats sont meilleurs. L'ensemble des configurations adaptées par BN se rapproche de la précision de référence. Ce phénomène est tout particulièrement marqué pour le modèle initialisé sur le tunnel piéton, qui présente une précision égale à 85% de la précision de référence. Quant au F_1-score, ce même modèle atteint également 85% de la valeur de référence.

La figure 6.22 met en regard certaines des prédictions des modèles évalués dans ce chapitre. On peut y voir que les modèles appris sur le tunnel piéton se traduisent par une forte tendance à la sous-détection. Lorsque l'apprentissage porte sur la base CODEBRIM, les résultats sont meilleurs, même si cer-

taines parties des anomalies ne sont pas détectées. De plus, le joint central ainsi que certaines zones à la texture granuleuse sont incorrectement prédits comme anomalie. Les modèles adaptés par sans supervision produisent tous deux un nombre prohibitif de fausses alarmes. Ils parviennent cependant à mieux détecter les fers apparents que le modèle appris sur le tunnel piéton. Les modèles obtenus par adaptation faiblement supervisée (ajustement exclusif des paramètres des couches de *Batch Normalization*) sont ceux qui génèrent le moins de fausses alarmes. De façon intéressante, on peut noter que les revêtements considérés à tort comme fer apparent pour les modèles appris sur la base CODEBRIM et adaptés depuis cette même base ont une apparence proche, ce qui tend à montrer que le processus d'adaptation faiblement supervisé conserve en partie les caractéristiques extraites par le modèle utilisé pour son initialisation.

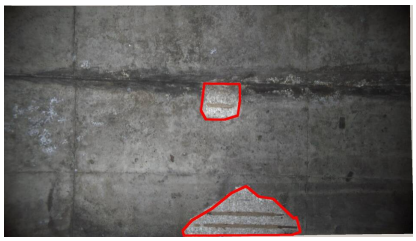


(a) Jeu intégral



(b) Jeu restreint

FIGURE 6.21 – Synthèse des performances sur Rive-de-Gier pour l'ensemble des modèles évalués sur ce domaine. Le marqueur (A) désigne un modèle qui a été adapté tandis que (S) indique un modèle utilisé sans adaptation. L'adaptation par BN emploie l'image décrite en figure 6.15.



Vérité terrain

(S) Rive-de-Gier (×1)

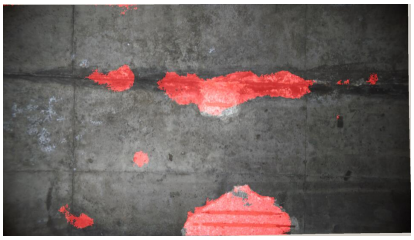
(S) Tunnel piéton, EP

(S) Tunnel piéton, F_1

(A) Sans supervision, EP

(A) Sans supervision, F_1

(S) CODEBRIM, EP et F_1



(A) BN, CODEBRIM

(A) BN, tunnel piéton

(A) BN, hasard

FIGURE 6.22 – Synthèse qualitative de l'ensemble des modèles évalués sur Rive-de-Gier. Chaque sous-titre correspond à la configuration dont est issue la prédiction. Le marqueur (A) désigne un modèle qui a été adapté tandis que (S) indique un modèle utilisé sans adaptation. L'adaptation par BN emploie l'image décrite en figure 6.15.

Conclusion du chapitre

Dans ce chapitre, nous avons mesuré l'influence du biais de domaine sur les modèles appris. De plus, nous avons évalué différentes stratégies pour prendre en compte ce biais de domaine et en réduire les conséquences sur la capacité de généralisation des modèles. Il en ressort les éléments suivants :

- La variabilité d'apparence entre ouvrages réduit considérablement les performances des réseaux convolutifs. Cela se traduit par un nombre conséquent de fausses alarmes et une proportion tout aussi importante d'anomalies non détectées.
- La méthode d'adaptation de domaine non supervisée que nous avons testée peine à fournir des résultats convaincants. En effet, même si les performances sont supérieures aux modèles non adaptés, de nombreuses fausses alarmes sont présentes. À travers la représentation t-SNE, nous avons pu observer que la méthode avait pourtant permis la création d'un espace de caractéristiques commun aux deux domaines. Or, nous avons constaté, par le biais d'exemples de prédictions, que certains éléments visuellement proches de fers apparents, ont été considérés à tort comme anomalie. Il est donc probable que, malgré une bonne concordance des deux domaines au sein de l'espace de caractéristiques considérés, les classes respectives de chacun des domaines ne soient pas alignées.
- L'ajout d'une faible part de supervision dans le processus d'adaptation, à travers l'ajustement exclusif des paramètres des couches de *batch normalization* à l'aide d'une image annotée du domaine cible, permet d'améliorer ces résultats et d'approcher ceux obtenus lorsque le modèle est directement appris sur ce même domaine. De plus, l'emploi d'un modèle appris pour la tâche d'intérêt comme modèle initial de cette méthode d'adaptation aboutit à l'apprentissage de meilleurs modèles, aussi bien sur le plan quantitatif (métriques) que qualitatif (exemples de prédictions, représentations t-SNE). Il convient toutefois de noter que cette méthode, testée sur un tunnel dans lequel les anomalies sont visuellement semblables, risque de présenter des performances moindres sur les ouvrages caractérisés par une plus forte variabilité visuelle.

Chapitre 7

Conclusion générale

7.1 Synthèse

Dans cette thèse, nous avons mis en œuvre plusieurs méthodes pour la cartographie des anomalies sur des structures de génie civil. Nous dressons une synthèse des contributions de notre travail dans cette section.

Banques de données Nous avons constitué et annoté, au pixel près, plusieurs banques d'images. Le travail d'annotation a été réalisé avec deux logiciels spécifiques développés par nos soins. Le premier logiciel permet d'entourer les anomalies avec des polygones et propose, de plus, d'alléger le travail d'annotation pour les données acquises en séquence. Pour ces dernières, lorsque deux images présentant un fort recouvrement sont à annoter, les anomalies définies sur la première image sont reportées sur la seconde après avoir été repositionnées par application d'une translation correspondant au décalage estimé entre les deux images. Ce décalage est déterminé par appariement de points caractéristiques (descripteurs SIFT). Le second logiciel d'annotation, dédié aux fissures, utilise une technique de relevé semi-automatique, utilisant le *fast-marching*, pour accélérer le processus de labellisation de ces dernières.

Pour les images photographiques, la banque d'images se compose de prises de vue portant sur différents ouvrages (entrées d'un tunnel piéton, un tunnel routier, un bâtiment universitaire, une trentaine de ponts). Dédiée à la reconnaissance de fers apparents, elle répertorie 2573 anomalies de ce type (certaines anomalies sont présentes plusieurs fois au sein des prises de vue, avec toutefois des conditions d'acquisitions différentes d'une image à l'autre). Notons que cette anomalie, pourtant fréquente sur les constructions en béton armé, est globalement peu considérée dans la littérature.

Concernant les relevés LCMS, cette banque est constituée d'une séquence de relevés acquis au sein d'un même tunnel routier et vise à la détection des fissures. Outre la préparation et l'annotation de cette séquence, nous avons implémenté deux prétraitements de la composante de profondeur, afin de prendre en compte la complexité de cette dernière (points hors-de-portée, adaptation

de la normalisation à la problématique de détection des fissures). Le premier consiste à améliorer la dynamique de la profondeur en modifiant la valeur des points hors-de-portée (profondeur centrée) tandis que le second vise à modéliser la surface du tunnel (profondeur ajustée). À notre connaissance, il n'existe pas d'autre base utilisant des relevés LCMS de voûtes et de piédroits de tunnels dont les fissures sont labellisées.

Apprentissage intra-domaine Les premières expérimentations ont cherché à évaluer la capacité des réseaux de neurones à prendre en compte la variabilité des anomalies au sein d'un même ouvrage pour deux types de cartographie, par quadrillage régulier et par segmentation sémantique.

Pour les images photographiques, nous montrons, pour la cartographie par quadrillage régulier, que les méthodes d'apprentissage profond peuvent s'avérer moins efficaces pour cette tâche que les autres approches d'apprentissage automatique (forêts aléatoires, *SVM*, etc.) si le jeu d'apprentissage est trop réduit ou insuffisamment représentatif du reste de l'ouvrage. Pour l'approche par segmentation sémantique, nous mettons au jour, à travers un apprentissage et une évaluation multi-échelles réalisée indépendamment sur trois sites, que les réseaux de neurones réussissent à modéliser la variabilité d'aspect, parfois importante, présente dans chacun des sites, atteignant des F_1-score allant de 47% à 73% selon le site considéré. Nous avons toutefois observé que le champ réceptif des modèles était parfois insuffisant pour détecter correctement certains fers apparents pris en plan rapproché au sein d'images fortement résolues. Le redimensionnement de ces images ou l'emploi d'une approche multi-échelles permettent de répondre en partie à cette problématique.

S'agissant des relevés LCMS, nous avons montré que la stratégie mono-grille (cartographie par quadrillage régulier) parvient difficilement à discerner les fissures lorsque ces dernières se situent sur le bord d'une sous-image. En effet, le voisinage de la fissure n'étant que partiellement disponible dans ce cas, le modèle peine à faire la différence entre une fissure excentrée et une simple aspérité du revêtement. Les stratégies multi-grilles (cartographie par quadrillages réguliers avec recouvrement partiel), à l'image de celle proposée par Cha *et al.* [9], permettent une meilleure prise en compte du voisinage des anomalies et accroissent ainsi les performances des modèles appris. Ces deux approches ont atteint des F_1-score de 54% pour la stratégie mono-grille et de 66% pour la stratégie multi-grilles. La cartographie par segmentation sémantique mise en œuvre a permis d'obtenir des relevés de fissures caractérisés par un θF_1-score de plus de 73% (sur le jeu de test restreint avec $\theta = 11$). Nous montrons, de plus, que l'utilisation combinée de la carte d'intensité et de la carte de profondeur (ajustée ou brute) représente la configuration qui permet d'atteindre les meilleurs résultats et ce, pour les deux types de cartographie.

Apprentissage inter-domaines Nous avons montré, à travers une étude multi-sites pour la segmentation d'images photographiques, que la différence d'aspects entre les différents sites constituait un problème critique. En effet,

lorsqu'un modèle est appris sur un premier ouvrage et évalué sur un second, on observe une chute de ses performances. Bien que cette baisse dépende du jeu d'apprentissage, laissant supposer que l'apprentissage sur certains de nos jeux permet d'atteindre une plus grande capacité de généralisation que l'apprentissage sur d'autres, elle demeure significative pour tous les jeux de données. En effet, selon le domaine considéré, les réseaux de neurones réussissent à conserver entre 20% et 90% du F_1-score obtenu par le modèle appris sur le même domaine (à l'échelle originale).

Plusieurs méthodes d'adaptation de domaine ont alors été mises en œuvre. Une première approche, non supervisée, proposait de construire, par autosupervision, un encodeur capable de plonger les deux domaines dans un espace de caractéristiques communs, avant d'y adjoindre un décodeur pour apprendre à l'architecture complète à réaliser la cartographie sur le jeu source. Si cette approche a permis d'améliorer le rappel, cette dernière métrique atteignant près de 43% (contre 7% pour le modèle non adapté), elle induit un plus grand nombre de fausses alarmes. En conséquence, la précision s'établit à 0,2% (contre 0,4% pour le modèle non adapté). Face à la faiblesse de ces résultats, nous avons développé puis évalué une méthode faiblement supervisée. Il s'agit de considérer un modèle appris exclusivement sur le jeu source, puis d'ajuster les poids de ses couches de *batch normalization* à l'aide d'une unique image annotée du jeu cible. Cette stratégie s'est avérée payante puisque nous sommes parvenus, avec le tunnel routier de Rive-de-Gier comme domaine cible, à conserver 85% du F_1-score obtenu par le modèle appris directement sur ce dernier tunnel.

7.2 Perspectives

Même si nos travaux ont permis l'investigation de plusieurs verrous scientifiques concernant l'application des méthodes de reconnaissance des formes à la problématique de la cartographie des anomalies, il reste un chemin conséquent à parcourir avant une utilisation de ces dernières en condition opérationnelle. Nous distinguons deux types de perspectives : les perspectives scientifiques, qui se concentrent sur les suites à donner aux travaux réalisés dans la thèse, et les perspectives opérationnelles, qui s'intéressent à la manière dont les algorithmes que nous avons mis en œuvre vont être mis au service de l'application.

7.2.1 Perspectives scientifiques

Banques de données L'obtention de données annotées et représentatives du domaine applicatif constitue un enjeu majeur dans le cadre des méthodes d'apprentissage automatique, aussi bien en ce qui concerne l'apprentissage des modèles que leur évaluation. Si la réalisation de campagnes d'acquisition de données permet de contrôler l'adéquation des données avec l'application, de telles campagnes sont coûteuses et complexes à organiser (mobilisation de plusieurs opérateurs, déplacement et assurance du matériel, nécessité de fermer le tunnel en question). De plus, l'annotation des données collectées, étape incontournable dans le cadre de l'apprentissage supervisé, représente un travail long

et fastidieux. Dans la thèse, les bases dédiées à la segmentation sémantique des anomalies (fers apparents pour les images photographiques et fissures pour les données LCMS) sont composées d'un nombre conséquent d'exemples. Ces derniers sont cependant loin de couvrir toute la variabilité d'aspect que l'on peut retrouver dans notre contexte opérationnel. Il conviendrait ainsi de travailler avec davantage de données.

Un premier levier, facilement actionnable, pour accroître la quantité d'exemples utilisés pour la segmentation sémantique des fers apparents réside dans l'usage de l'intégralité de la base CODEBRIM [40]. En effet, lors de l'adaptation de cette base à notre protocole expérimental, près de 77% des images en ont été écartées en raison de l'absence de fers apparents au sein de ces dernières. Pour ce faire, une architecture à deux branches pourrait être employée. La première branche reproduirait les travaux des auteurs de la base (*i.e.* reconnaissance, par classification, de 5 types d'anomalies : fissure, épaufrure, efflorescence, fers apparents, corrosion) tandis que l'autre demeurerait dédiée à la segmentation sémantique des fers apparents. Placé en début de la chaîne neuronale, l'encodeur serait alors appris grâce à l'intégralité des exemples, ce qui bénéficierait alors aux deux branches qui lui succèdent. En particulier, l'apprentissage de la branche réalisant la segmentation sémantique des fers apparents s'en verrait renforcé. La figure 7.1 résume cette méthode.

Une deuxième option, ayant également un coût opérationnel faible, consiste à utiliser l'intégralité des données LCMS déjà acquises par le groupe ENDSUM du Cerema. En effet, dans la thèse, les données LCMS que nous avons utilisées ne proviennent que d'un seul tunnel, alors que deux autres tunnels ont aussi été acquis par le Cerema avec le capteur LCMS et sont donc disponibles. Il restera cependant à annoter l'ensemble de ces données, ce qui représente possiblement plusieurs semaines de travail avec les outils actuellement développés.

Une autre approche, utilisées par certaines contributions de notre domaine applicatif [95, 131], consiste à récupérer des images *via* les moteurs de recherche en indiquant pour requête le nom de l'anomalie considérée, la requête devenant ensuite la vérité terrain attribuée à l'ensemble des images ainsi collectées. On parle d'approche *webly supervised* [157]. Contrairement à la précédente stratégie, cette approche permet d'obtenir une importante quantité de données, au prix d'une grande hétérogénéité au sein des images récupérées, aussi bien en termes d'aspect visuel que de la concordance entre la requête transmise au moteur de recherche et la vérité terrain réelle de ces images.

Soulignons enfin le nombre croissant de bases d'images annotées publiquement disponibles et dédiés à la détection des anomalies, principalement des fissures, sur des structures de génie civil. Bianchi *et al.* [158] répertorient un certain nombre d'entre elles. Même si ces bases présentent l'intérêt d'être rapidement utilisables, il convient de noter qu'une vérification systématique de chaque exemple ainsi que de l'annotation qui lui est associée demeure nécessaire. La consultation des exemples a pour but de jauger la représentativité des exemples par rapport à notre domaine d'application et celle des annotations sert à détecter la présence d'erreurs de labellisation. À titre d'exemple, en travaillant sur la base CODEBRIM, nous avons pu constater la présence d'une

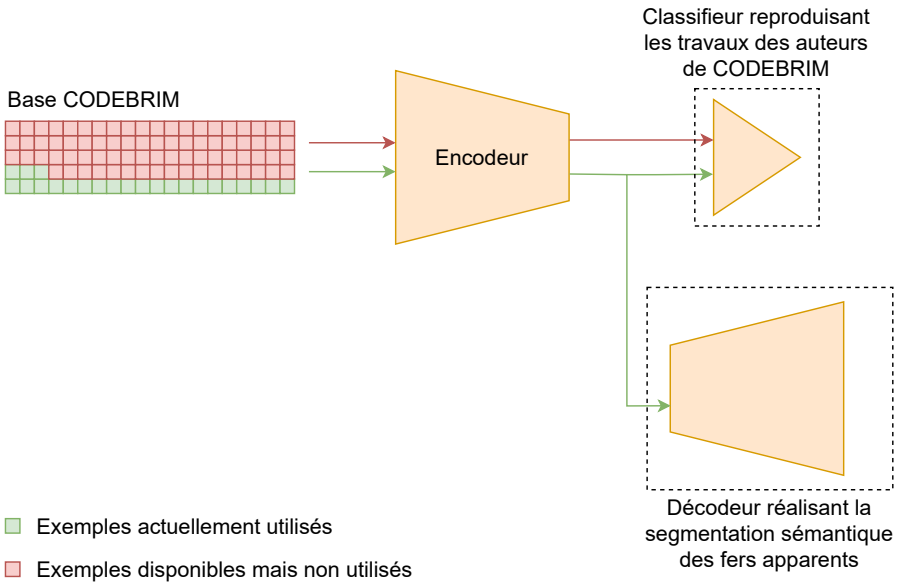


FIGURE 7.1 – Approche envisageable pour utiliser l'ensemble des données de la base CODEBRIM. Chaque carré représente 1% des images de cette base. Les flèches indiquent le chemin emprunté par les données (flèches vertes pour les exemples utilisés dans la thèse et rouge pour les autres).

dizaine d'erreurs de ce type dans les annotations.

Outre la quantité et la représentativité des exemples employés dans l'apprentissage et l'évaluation de nos modèles, la prise en compte d'une plus grande diversité d'anomalies représente également une extension possible de la thèse. En effet, sur les 47 types d'anomalies recensées par le CETU [2], seules quatre d'entre eux sont présents dans nos expérimentations. Bien que les anomalies que nous considérons dans la thèse comptent parmi les plus importantes aux yeux des inspecteurs (à l'image des fissures, par exemple), il conviendrait d'enrichir nos banques de données pour y inclure davantage d'anomalies et, ainsi, apprendre des modèles capables de détecter une plus grande variété d'anomalies. En guise de première approche, nous avons exclusivement considéré des modèles binaires dans la thèse. Il serait toutefois pertinent d'avoir, *in fine*, un modèle multi-anomalies pour chaque type de données (LCMS/photographies). Un tel modèle présenterait alors un temps d'analyse plus faible que celui qu'aurait un comité de modèles binaires puisqu'une seule inférence du modèle multi-anomalies serait nécessaire pour créer une cartographie pour toutes les classes d'anomalies considérées, là où le comité demanderait autant d'inférences que de classes d'anomalies.

Modélisation de la surface du tunnel (données LCMS) Concernant la profondeur, il serait intéressant de mettre en œuvre une modélisation de la

surface du tunnel agissant à plus grande échelle qu'à celle de la sous-image, afin de mieux distinguer les équipements, situés au-devant de la paroi, des cavités, situées en retrait. Dans le cadre d'un projet de fin d'études [38, 159] réalisé au sein du groupe ENDSUM du Cerema, une approche basée sur les B-splines, somme pondérée de fonctions splines (polynomiales), est implémentée et comparée à la méthode de régression robuste que nous avons employée. Ces splines sont définies à partir d'un ensemble de noeuds positionnés sur la surface qui permet d'ajuster le modèle localement. Il en ressort que l'utilisation de ces surfaces permet de modéliser la courbure globale du tunnel tout en s'ajustant aux variations locales, contrairement à un modèle polynomial classique. En contrepartie de sa robustesse, la méthode de modélisation globale présentée dans [38] requiert de définir manuellement les nœuds à partir desquels les B-splines sont construites. Ces travaux ont ensuite été poursuivis et appliqués à la problématique du relevé des équipements au sein des tunnels [160]. En plus de faciliter le travail de documentation de la structure, ce qui représente un enjeu opérationnel en tant que tel, l'identification des équipements peut également contribuer à la cartographie des anomalies. En effet, si les équipements sont connus et localisés au sein de l'ouvrage, il devient possible de neutraliser les zones où ils se situent dans la cartographie, c'est-à-dire de considérer que ces zones correspondent à des revêtements sains et ce, indépendamment des prédictions faites par le modèle de cartographie considéré. Notons toutefois que l'apprentissage des équipements présente des problématiques analogues à celles que nous rencontrons lors de l'apprentissage des anomalies (nécessité de constituer de larges banques de données annotées, variabilité d'aspects des équipements, etc.).

Apprentissage intra-domaine Pour l'apprentissage sur les données LCMS, une stratégie susceptible d'améliorer les performances des modèles réside dans la transformation des annotations de fissures en cartes de distance [161, 104]. Cette opération permet de traiter efficacement le déséquilibre entre classes puisque l'on passe d'une information binaire (fissure/non fissure), à une information continue (distance à la fissure la plus proche).

Pour les images photographiques, tirer profit du recouvrement entre images pour l'apprentissage et la prédiction, à l'image des travaux de Schmugge *et al.* [114], pourrait renforcer la robustesse des prédictions des réseaux de neurones, tout en posant les premiers jalons de la construction de la cartographie globale.

De façon plus générale, les modèles utilisés demeurent rudimentaires, l'emploi d'architectures et de méthodes plus sophistiquées est susceptible d'améliorer les résultats. En particulier, il serait intéressant de mesurer l'apport des modèles gérant nativement les aspects multi-échelles [162].

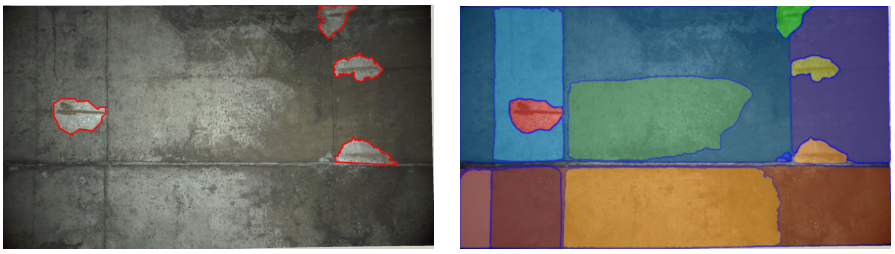
Apprentissage inter-domaines Lors de l'expérimentation sur l'apprentissage intra-domaine, nous avons constaté que les modèles multi-échelles présentaient de meilleurs résultats que les modèles mono-échelles pour les images non redimensionnées. Leur utilisation dans ce contexte d'évaluation inter-domaines

pourrait réduire l'influence du biais de domaine sur les performances des modèles, en particulier lorsque les anomalies admettent des tailles apparentes différentes d'un domaine à l'autre. Dans cette même optique, nous avons constaté que les modèles reposant sur des cartes de caractéristiques préalablement extraites des données semblaient moins sensibles au biais de domaine que les réseaux de neurones opérant directement sur les données. Même si cette stratégie tend à se raréfier dans la littérature, l'adjonction de cartes de caractéristiques aux données brutes, telle que proposée par Makantasis *et al.* [93], pourrait être intéressante à expérimenter.

En ce qui concerne les méthodes d'adaptation de domaine, la question de l'efficacité des approches autosupervisées et, plus généralement, des approches non supervisées dans notre champ applicatif reste ouverte. En effet, bien que l'expérimentation que nous avons conduite n'a pas été concluante, la littérature traitant de l'adaptation de domaine non supervisée est vaste et s'enrichit continuellement [163, 164]. Les stratégies autosupervisées bénéficient d'un engouement du même ordre [165, 166, 153].

S'agissant de la méthode d'adaptation de domaine faiblement supervisée que nous avons mise en œuvre au chapitre 6, il serait pertinent de poursuivre son évaluation. En effet, cette approche n'a été appliquée que sur un seul site. Même si les résultats sont encourageants, on ne peut pas, à ce stade, conclure sur son efficacité dans un cadre plus général. Il est donc nécessaire de réitérer ce procédé sur d'autres ouvrages et, également, d'autres type d'anomalies. Par ailleurs, il serait intéressant de mesurer l'évolution des performances des modèles adaptés lorsque le processus d'adaptation porte sur plusieurs images annotées.

Stratégies non supervisées De façon générale, il convient de souligner que l'ensemble des approches développées, qu'elles concernent l'apprentissage intra- ou inter-domaines, peut être complété par des stratégies non supervisées. Par exemple, au printemps 2023, *FAIR* (Meta) a publié *SAM* [167] (pour *Segment Anything Model*), un modèle capable de réaliser une segmentation non sémantique sur une image (*i.e.* un découpage de l'image en zones visuellement cohérentes sans pour autant donner lieu à une attribution de label à chacune des zones). La figure 7.2 présente le résultat de cet algorithme sur une image du tunnel de Rive-de-Gier. On constate que les quatre fers apparents visibles dans l'image figurent parmi les zones prédites par le modèle et sont, de plus, parfaitement délimités. Grâce à un tel découpage de l'image, il devient possible de guider un modèle de segmentation sémantique afin de détecter les fers apparents. En effet, on peut, une fois ce dernier modèle appris, réaliser un vote majoritaire parmi ses prédictions au sein de chaque composante connexe définie par *SAM* pour « binariser » les prédictions (chaque composante connexe se voit attribuer une unique prédiction). Ce faisant, l'influence des fausses alarmes situées au voisinage des anomalies s'en voit réduite. En effet, nous avons constaté, tout au long de nos expérimentations, que les modèles avaient des difficultés à délimiter précisément les anomalies, de nombreuses prédictions débordant les défauts et



(a) Vérité terrain (b) Résultat

FIGURE 7.2 – Illustration de la méthode de segmentation (non sémantique) *SAM*. Les fers apparents sont détourés en rouge dans la vérité terrain (a) et l'ensemble des zones prédites par *SAM* est en surbrillance dans l'image de résultat (b).

certaines autres ne couvrant que partiellement ces derniers. L'introduction du vote majoritaire peut donc atténuer ce phénomène.

Par ailleurs, puisque le modèle SAM est publiquement disponible, aussi bien en ce qui concerne le code écrit par les équipes de FAIR que les poids du modèle appris, il peut également être intégré à nos logiciels d'annotation pour simplifier le processus de labellisation. Certaines plateformes en ligne, à l'image de Roboflow [1], proposent déjà des outils d'annotation reposant sur SAM.

7.2.2 Perspectives opérationnelles

Intégration logicielle Pour que l'ensemble des approches puissent être utilisables par un opérateur non expert de l'apprentissage automatique, il conviendrait d'intégrer ces approches au sein d'un logiciel pour en faciliter la mise en œuvre. Plus qu'un simple support à ces approches, ce programme aurait également vocation à permettre à l'utilisateur d'interagir avec ces derniers. En particulier, la possibilité lui serait donnée de corriger les cartographies produites en ajoutant manuellement des relevés d'anomalie ou, à l'inverse, de supprimer les zones qu'il estimera prédites à tort comme anomalies. Par ailleurs, ce logiciel devrait aussi être capable de « fusionner » les différentes cartographies locales produites pour générer une cartographie globale de l'ouvrage, algorithme qu'il reste à concevoir. Notons qu'un logiciel de visualisation des données des tunnels a été développé au sein du groupe ENDSUM en 2015 [30] et pourrait servir de base pour l'intégration des stratégies de détection que nous avons développées dans la thèse.

Évaluation opérationnelle des modèles Durant la thèse, nous avons évalué nos modèles grâce à plusieurs métriques. Si cette évaluation a permis de positionner les approches les unes par rapport aux autres, nous ne disposons pas

1. `https://roboflow.com/`, consulté le 07 octobre 2023.

de critère permettant d'évaluer de façon absolue l'intérêt opérationnel d'un modèle, d'autant plus que certaines des métriques employées tendent à admettre des comportements antagonistes. Par exemple, nous avons pu constater qu'en cas de fort déséquilibre entre classes, les deux critères de sélection du modèle (exactitude pondérée et F_1-score) donnent généralement lieu à des compromis rappel/précision différents, le critère fondé sur l'exactitude pondérée ayant tendance à favoriser le rappel au détriment de la précision et inversement pour le critère reposant sur le F_1-score. Une question naturelle est alors de savoir laquelle de ces deux situations est préférable du point de vue des inspecteurs. D'un côté, avoir un rappel élevé et une précision faible permet à ces derniers de détecter l'essentiel des anomalies des ouvrages mais leur demande, en contrepartie, de retirer un grand nombre de fausses alarmes. De l'autre côté, prioriser la précision sur le rappel réduit la charge de travail dédiée au filtrage des fausses alarmes mais induit alors un risque important que certaines anomalies ne soient pas détectées. Déterminer où positionner le curseur n'est pas simple et la réponse dépendra en partie de l'interface qui sera proposée aux utilisateurs (selon, par exemple, la facilité que présentera cette dernière pour supprimer les fausses alarmes produites par les modèles).

Plus généralement, la détermination des critères pour évaluer l'utilité opérationnelle des modèles pourrait passer par la réalisation de sondages auprès des inspecteurs. Sur une même base d'exemples de prédictions hétérogènes présentées à un panel d'inspecteurs, il serait demandé à chacun d'eux d'attribuer une note à chaque prédiction. Outre cette appréciation, il pourrait aussi être proposé aux inspecteurs de corriger ces prédictions à l'aide du logiciel mentionné au paragraphe précédent et de mesurer le temps nécessaire à la réalisation de l'opération. Une fois les données recueillies, une première approche pourrait être de calculer la moyenne de ces deux indicateurs pour chaque exemple de prédiction puis de construire un modèle de régression linéaire entre ces indicateurs moyens et les métriques associées à la prédiction correspondante.

Une fois capables d'évaluer l'intérêt opérationnel d'un modèle, nous pourrons alors comparer l'intérêt des données LCMS par rapport aux images photographiques.

Prise en compte d'autres verrous Nous n'avons étudié que deux verrous opérationnels sur les quatre identifiés en introduction de ce manuscrit. Ainsi, les problématiques de rareté de certaines anomalies et de recouvrement entre classes n'ont pas été abordées.

La rareté de certaines anomalies peut être appréhendée de différentes manières. Pour les anomalies rares caractérisées par une faible variabilité d'aspect, une petite quantité d'exemples peut suffire à construire une base représentative de l'anomalie en question. Dès lors, il est possible d'employer des méthodes spécifiquement dédiées à l'apprentissage de réseaux de neurones à partir de bases réduites. On parle de « *few-shot learning* ». Une telle approche a, par exemple, été expérimentée par Lin *et al.* [168] pour la détection d'anomalies dans le cadre du contrôle qualité de chaînes de production. Dans le cas où il

n'est pas possible de constituer une base répondant à ce critère de représentativité, cette approche n'est plus applicable. Les méthodes d'apprentissage à une classe représentent alors une alternative. Dans cette configuration, on formule le problème dans l'autre sens : plutôt que de chercher explicitement à détecter les anomalies rares pour lesquelles on ne dispose que d'une quantité limitée de données, on va, à l'inverse, modéliser la distribution de la classe des revêtements sains et considérer comme anomalie tout élément visuel qui s'écarte trop de cette distribution. Ce faisant, la rareté des anomalies n'est plus une difficulté mais la méthode tend alors à générer davantage de fausses alarmes. Faula *et al.* [169] ont, par exemple, développé une telle approche dans notre champ applicatif.

Pour la problématique du recouvrement entre classes, la situation est plus délicate. S'il existe bien des méthodes pour tenir compte d'annotations empreintes d'erreurs ou d'approximations [170, 171], elles ne peuvent pas être étendues à l'indécidabilité de toute une classe d'anomalie. Une première piste consiste à multiplier les modalités d'acquisition afin d'augmenter la probabilité que l'anomalie soit clairement distinguable au sein de l'une d'elles (les zones humides peuvent potentiellement apparaître dans les images infrarouges, les pertes de matières sont visibles dans les relevés laser). Une seconde piste est de réaliser plusieurs acquisitions d'un même site à plusieurs mois d'intervalle. L'évolution de certaines zones pourrait révéler la présence de certaines anomalies telles les zones humides dont les dépôts sédimentaires diffèreraient d'une captation à l'autre.

Publications de l'auteur

Conférences internationales avec comité de lecture

G. Decor, M. D. Bah, P. Foucher, P. Charbonnier et F. Heitz, « Defect Detection in Tunnel Images using Random Forests and Deep Learning », *dans International Conference on Pattern Recognition Systems (ICPRS)*, (Tours, France), p. 1–6, 2019.

P. Foucher, G. Decor, F. Bock, P. Charbonnier et F. Heitz, « Evaluating of a Deep Learning Method for Detecting Exposed Bars From Images », *dans NSG2021 2nd Conference on Geophysics for Infrastructure Planning, Monitoring and BIM*, (Bordeaux, France), p. 1–5, European Association of Geoscientists & Engineers, 2021.

Conférences nationales avec comité de lecture

G. Decor, M. D. Bah, P. Foucher, P. Charbonnier et F. Heitz, « Détection d'anomalies dans des images de tunnels par forêts d'arbres aléatoires ou par apprentissage profond », *dans Congrès Jeunes Chercheurs ORASIS*, (Saint-Dié-des-Vosges, France), p. 1–8, 2019.

G. Decor, P. Foucher, P. Charbonnier et F. Heitz, « IronNet : détection de fers apparents au sein de structures en béton armé », *dans Congrès national sur la Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP)*, p. 1–3, 2020.

Communications

G. Decor, P. Foucher, P. Charbonnier et F. Heitz, « Poster pour la présentation du sujet de thèse », *dans Journée Machine Learning du Cerema*, (Marne-la-Vallée, France), p. 1–3, Cerema, 2018.

G. Decor, P. Foucher, P. Charbonnier et F. Heitz, « Poster pour la présentation de l'avancement des travaux de thèse », *dans Journée de rentrée des doctorants de l'ED269*, (Illkirch-Graffenstaden, France), 2019.

Annexe A

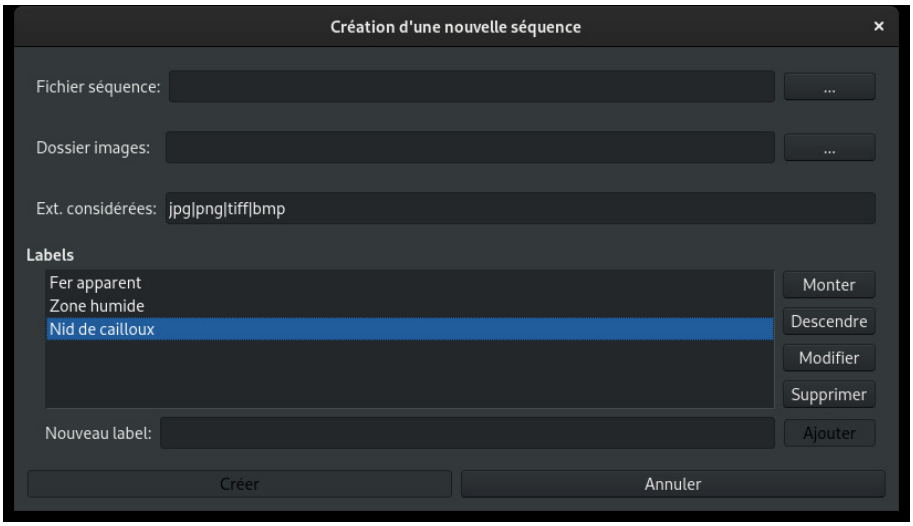
Logiciels d'annotation

Deux logiciels d'annotation ont été développés pour créer les vérités terrain des différents jeux de données. Le premier permet d'effectuer des relevés sous la forme de polygones englobants et a été utilisé pour l'annotation des fers apparents. Le second intègre une méthode de *fast-marching* pour faciliter le relever des fissures. Voir chapitre 2 pour plus d'informations.

A.1 Logiciel d'annotation pour les fers apparents

Fonctionnalités :

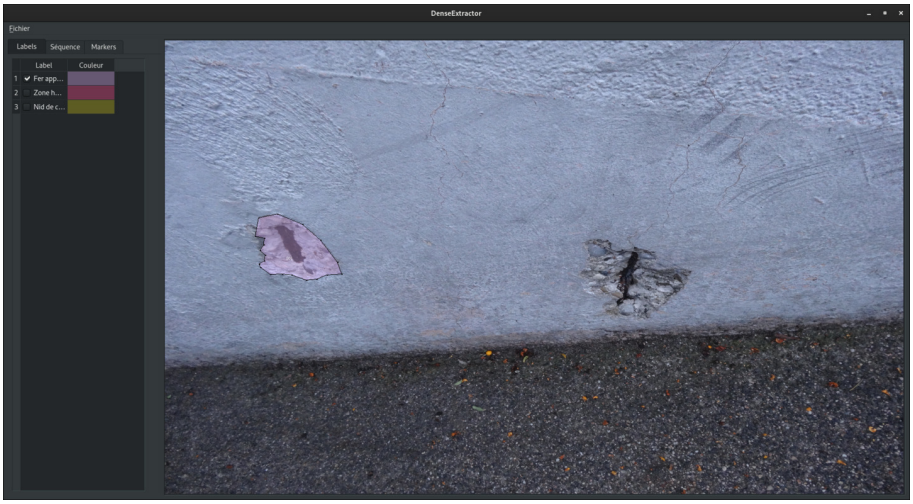
- Création de « projets d'annotation » (*i.e.* fichier encapsulant les données et les vérités terrain) ;
- Annotation par polygones englobants ;
- Annotation de plusieurs type d'anomalies ;
- Modification/suppression d'annotations précédemment définies ;
- Affichage des polygones en surimpression de l'image ; possibilité de masquer certains polygones dans l'interface ;
- Fusion de polygones d'intersection non vide et représentant la même anomalie ;
- Dans les séquences acquises linéairement, calcul automatique du déplacement entre deux images successives puis translation des annotations ;
- Matricialisation des polygones pour générer des vérités terrain adaptées à la segmentation sémantique (*i.e.* définies au pixel près).



Menu de création d'un projet d'annotation de séquence

Interface principale. L'image centrale permet de définir les polygones englobants par sélection de points à la souris et validation au clavier (touche A pour valider un polygone). Le volet de gauche indique les labels disponibles et le label actuellement attribué aux polygones nouvellement créés. En double-cliquant sur la case colorée à droite de l'intitulé des labels, il est possible de changer la couleur associée au label.

Le volet de gauche indique la liste des images à annoter. Le nom de l'image actuellement affichée dans la partie centrale est sélectionné dans la liste. En validant les annotations d'une image (touche V), on charge automatiquement l'image suivante (avec les annotations associées le cas échéant). Il est également possible de choisir l'image à annoter en double-cliquant sur le nom de l'image dans le volet de gauche.



Sur cet exemple, le fer apparent de gauche a été relevé. L'intérieur du polygone associé apparaît alors en surbrillance selon la couleur choisie par l'utilisateur.

Dans le volet de gauche, on voit la liste des marqueurs (*e.g.* anomalies relevées). Il est possible de supprimer ou de masquer tout ou partie des marqueurs.

A.2 Logiciel d'annotation pour les fissures

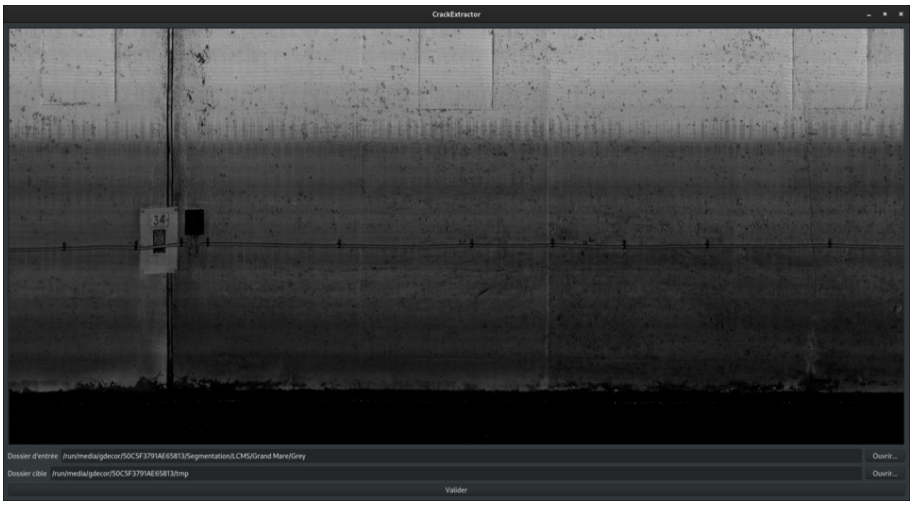
Fonctionnalités :

— Annotation de fissures au pixel près ;

— Relevé guidé par un algorithme de *fast-marching* ;

— Modification/suppression d'annotations précédemment définies ;

— Enregistrement de la vérité terrain sous forme d'image en noir et blanc (un pixel noir représente un revêtement sain et un pixel blanc une fissure).



Interface principale. Pour relever une fissure, il faut cliquer sur l'image au niveau des deux extrémités de cette dernière. L'algorithme de *fast-marching* en propose alors un tracé.

La fissure présente sur la droite de l'image a été annotée. Le relevé est matérialisé par des pixels noirs positionnés sur la fissure labellisée.

Bibliographie

[1] L. van der Maaten et G. E. Hinton, « Visualizing High-Dimensional Data Using t-SNE », *Journal of Machine Learning Research*, vol. 9, p. 2579–2605, 2008.

[2] F. Spataro, C. Boulogne, V. Robert, S. Frachon, C. Larive, J. Kasperski, Y. Peru, D. Subrin et A. Robert, « Guide le linspection des tunnels routiers », rap. tech., Centre d'Étude des Tunnels (CETU), Bron, 2015.

[3] P. Foucher, P. Charbonnier, T. Noël, Y. Fosse et J.-F. Hebert, « Scanning Tunnels with Two Very High-Resolution Laser Devices and a Stacker », *in Optical 3D Metrology (O3DM)*, vol. XLII-2/W18, (Strasbourg, France), p. 39–46, 2019.

[4] K. He, X. Zhang, S. Ren et J. Sun, « Deep Residual Learning for Image Recognition », *in Conference on Computer Vision and Pattern Recognition (CVPR)*, (Las Vegas, NV, USA), p. 770–778, IEEE, 2016.

[5] O. Ronneberger, P. Fischer et T. Brox, « U-Net : Convolutional Networks for Biomedical Image Segmentation », *in Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, p. 234–241, Munich, Germany : Navab, Nassir and Hornegger, Joachim and Wells, William M. and Frangi, Alejandro F., springer international publishing éd., 2015.

[6] V. Badrinarayanan, A. Kendall et R. Cipolla, « SegNet : A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation », *Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, p. 2481–2495, 2017.

[7] Y. Zhang, D. Sidibé, O. Morel et F. Mériaudeau, « Deep multimodal fusion for semantic image segmentation : A survey », *Image and Vision Computing*, vol. 105, p. 104042, 2021.

[8] S. Sinha et P. Fieguth, « Segmentation of buried concrete pipe images », *Automation in Construction*, vol. 15, no. 1, p. 47–57, 2006.

[9] Y.-J. Cha, W. Choi et O. Büyüköztürk, « Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks », *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, p. 361–378, 2017.

[10] Y. Fei, K. C. P. Wang, A. Zhang, C. Chen, J. Q. Li, Y. Liu, G. Yang et B. Li, « Pixel-Level Cracking Detection on 3D Asphalt Pavement Images Through Deep-Learning- Based CrackNet-V », *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, p. 273–284, 2019.

[11] L. Xu, S. Lv, Y. Deng et X. Li, « A Weakly Supervised Surface Defect Detection Based on Convolutional Neural Network », *IEEE Access*, vol. 8, p. 42285–42296, 2020.

[12] L. Duan, H. Geng, J. Pang et J. Zeng, « Unsupervised Pixel-level Crack Detection Based on Generative Adversarial Network », *in International Conference on Multimedia Systems and Signal Processing (ICMSSP)*, ICMSSP 2020, (New York, NY, USA), p. 6–10, Association for Computing Machinery, 2020.

[13] K. Zhang, Y. Zhang et H. D. Cheng, « Self-Supervised Structure Learning for Crack Detection Based on Cycle-Consistent Generative Adversarial Networks », *Journal of Computing in Civil Engineering*, vol. 34, no. 3, p. 04020004, 2020.

[14] W.-C. Hung, Y.-H. Tsai, Y.-T. Liou, Y.-Y. Lin et M.-H. Yang, « Adversarial Learning for Semi-Supervised Semantic Segmentation », *in British Machine Vision Conference (BMVC)*, (Newcastle, England), p. 1–12, BMVA Press, 2018.

[15] G. Li, J. Wan, S. He, Q. Liu et B. Ma, « Semi-Supervised Semantic Segmentation Using Adversarial Learning for Pavement Crack Detection », *IEEE Access*, vol. 8, p. 51446–51459, 2020.

[16] S. Shim, J. Kim, G.-C. Cho et S.-W. Lee, « Multiscale and Adversarial Learning-Based Semi-Supervised Semantic Segmentation Approach for Crack Detection in Concrete Structures », *IEEE Access*, vol. 8, p. 170939–170950, 2020.

[17] Y. Liu et J. K. W. Yeoh, « Vision-Based Semi-Supervised Learning Method for Concrete Crack Detection », *in Construction Research Congress*, p. 527–536, 2020.

[18] K. Simonyan et A. Zisserman, « Very Deep Convolutional Networks For Large-Scale Image Recognition », *in International Conference on Learning Representations (ICLR)*, (San Diego, California, USA), p. 1–14, 2015.

[19] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh et D. Batra, « Grad-CAM : Visual Explanations from Deep Networks via Gradient-Based Localization », *in International Conference on Computer Vision (ICCV)*, (Venice, Italy), p. 618–626, IEEE, 2017.

[20] T. Chen, S. Kornblith, M. Norouzi et G. Hinton, « A Simple Framework for Contrastive Learning of Visual Representations », *in International Conference on Machine Learning (ICML)*, p. 1597–1607, PMLR, 2020.

[21] Y. Sun, E. Tzeng, T. Darrell et A. A. Efros, « Unsupervised Domain Adaptation through Self-Supervision », *arXiv preprint*, no. arXiv :1909.11825, p. 1–16, 2019.

[22] « Répartition des tunnels routiers français par itinéraire et nombre de tubes », rap. tech., Observatoire des tunnels, Centre d'Étude des Tunnels (CETU), 2021.

[23] « Liste des tunnels du Réseau Ferré National », fichier de données CSV, Plateforme data.gouv.fr, 2021.

[24] E. Moisan, *Imagerie 3D du "tube entier" des tunnels navigables*. Thèse de doctorat, Université de Strasbourg, Strasbourg, 2017.

[25] Y. Liu, J. Yao, X. Lu, R. Xie et L. Li, « DeepCrack : A deep hierarchical feature learning architecture for crack segmentation », *Neurocomputing*, p. 139–153, 2019.

[26] P. Charbonnier, P. Foucher, P. Chavant, V. Muzet, D. Prybyla, T. Perrin, J.-L. Albert, P. Grussenmeyer, M. Koehl et S. Guillemin, « An image-based inspection system for canal-tunnel heritage », *International Journal of Heritage in the Digital Era*, vol. 3, no. 1, p. 197–217, 2014.

[27] A. Guittet, « Repérage d'un système d'inspection dans un tunnel-canal », projet de fin d'études, Institut National des Sciences Appliquées (INSA), Strasbourg, 2012.

[28] P. Chavent, « Évaluation absolue de méthodes de localisation et de reconstruction panoramique et photogrammétrique dun tunnel à partir dun nuage de points de référence », projet de fin d'études, Institut National des Sciences Appliquées (INSA), Strasbourg, 2013.

[29] B. Champier, « Bathymétrie en labsence de signal GPS : application aux canaux urbains et aux tunnels canaux », projet de fin d'études, Institut National des Sciences Appliquées (INSA), Strasbourg, 2014.

[30] M. Clog, « Développement dun logiciel de visualisation et dexploitation de séquences dimages de voûte des tunnels », stage de fin d'études, Université de Strasbourg, Strasbourg, 2015.

[31] G. Decor, « Exploration des Méthodes de Deep Learning pour l'Analyse d'Images de Tunnels », mémoire de Master, Université de Strasbourg, 2018.

[32] M. D. Bah, « Détection automatique des défauts et fissures sur les tunnels navigables et routiers par analyse d'images », mémoire de Master, Université de Toulouse, 2015. Spécialité de Master : Télédétection.

[33] P. Foucher, M. D. Bah, P. Charbonnier, C. Boulogne et C. Larive, « Classification automatique de défauts sur des images de tunnels par forêts d'arbres aléatoires », *in Congrès national sur la Reconnaissance de*

Formes et l'Intelligence Artificielle (RFIA), (Clermont-Ferrand, France), p. 1–2, 2016.

[34] L. Breiman, « Random Forests », *Machine Learning*, vol. 45, no. 1, p. 5–32, 2001.

[35] « Site de Pavetmetrics ». www.pavemetrics.com.

[36] P. Gayte, H. Guiraud, P. Rossigny, F. Palhol et E. Delaval, « IQRN 3D : A tool for better management of French road assets », *in 26th World Road Congress*, (Abu Dhabi , United Arab Emirates), p. 10, 2019.

[37] A. Carreaud, « Comment bien faire la peau dun tunnel ? », projet de Recherche Technologique (PRT), Institut National des Sciences Appliquées (INSA), Strasbourg, 2020.

[38] M. Tual, « Imagerie 3D haute résolution pour l'inspection des tunnels », projet de fin d'études, Université de Strasbourg, Strasbourg, France, 2021.

[39] S. Biderman et W. J. Scheirer, « Pitfalls in Machine Learning Research : Reexamining the Development Cycle », *in NeurIPS*, (Conférence Virtuelle), p. 12, 2020.

[40] M. Mundt, S. Majumder, S. Murali, P. Panetsos et V. Ramesh, « Meta-Learning Convolutional Neural Architectures for Multi-Target Concrete Defect Classification With the COncrete DEfect BRidge IMage Dataset », *in Conference on Computer Vision and Pattern Recognition (CVPR)*, (Long Beach, California, USA), p. 11196–11205, 2019.

[41] Y. LeCun, Y. Bengio et G. E. Hinton, « Deep learning », *Nature*, vol. 521, p. 436–444, 2015.

[42] A. Cauchy, « Méthode générale pour la résolution des systèmes déquations simultanées », *Comptes Rendus de lAcadémie des Sciences de Paris*, vol. 25, p. 536–538, 1847.

[43] G. Decor, M. D. Bah, P. Foucher, P. Charbonnier et F. Heitz, « Defect Detection in Tunnel Images using Random Forests and Deep Learning », *in International Conference on Pattern Recognition Systems (ICPRS)*, (Tours, France), p. 1–6, 2019.

[44] D. Lowe, « Object recognition from local scale-invariant features », *in Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, p. 1150–1157 vol.2, 1999.

[45] M. A. U. Khan, D. Nazir, A. Pagani, H. Mokayed, M. Liwicki, D. Stricker et M. Z. Afzal, « A Comprehensive Survey of Depth Completion Approaches », *Sensors*, vol. 22, no. 18, p. 6969, 2022.

[46] J. Hu, C. Bao, M. Ozay, C. Fan, Q. Gao, H. Liu et T. L. Lam, « Deep Depth Completion from Extremely Sparse Data : A Survey », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–20, 2022. Conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence.

[47] P. J. Huber, *Robust Statistics*. Berlin, Heidelberg : Springer, 1981.

[48] P. Charbonnier, *Modèles de forme et d'apparence en traitement d'images*. Habilitation à diriger des recherches, Université de Strasbourg, 2009.

[49] P. W. Holland et R. E. Welsch, « Robust regression using iteratively reweighted least-squares », *Communications in Statistics - Theory and Methods*, vol. 6, no. 9, p. 813–827, 1977.

[50] L. Cohen et R. Kimmel, « Global minimum for active contour models : a minimal path approach », *in Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 666–673, 1996.

[51] P. Charbonnier et J.-M. Moliard, « Calculs de chemins minimaux, suivi de fissures et autres applications », *in Journées des Sciences de l'Ingénieur du réseau des laboratoires des Ponts et Chaussées*, (Dourdan, France), p. 201–206, 2003.

[52] J. A. Sethian, « A fast marching level set method for monotonically advancing fronts », *Proceedings of the National Academy of Sciences*, vol. 93, no. 4, p. 1591–1595, 1996.

[53] W. S. McCulloch et W. Pitts, « A logical calculus of the ideas immanent in nervous activity », *The bulletin of mathematical biophysics*, vol. 5, no. 4, p. 115–133, 1943.

[54] F. Rosenblatt, « The Perceptron : A Perceiving and Recognizing Automaton », Rap. tech. 85-460-1, Cornell Aeronautical Laboratory, New York, 1957.

[55] M. Minsky et S. Papert, *Perceptrons : An Introduction to Computational Geometry*. MIT Press, 1969.

[56] A. Krizhevsky, I. Sutskever et G. E. Hinton, « ImageNet Classification with Deep Convolutional Neural Networks », *in International Conference on Neural Information Processing Systems*, vol. 1*in NIPS'12*, (Lake Tahoe, Nevada), p. 1097–1105, Curran Associates Inc., 2012.

[57] X. Glorot, A. Bordes et Y. Bengio, « Deep Sparse Rectifier Neural Networks », *in International Conference on Artificial Intelligence and Statistics*, vol. 15*in Proceedings of Machine Learning Research*, (Fort Lauderdale, FL, USA), p. 315–323, PMLR, 2011.

[58] G. Cybenko, « Approximation by superpositions of a sigmoidal function », *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, p. 303–314, 1989.

[59] Z. Lu, H. Pu, F. Wang, Z. Hu et L. Wang, « The expressive power of neural networks : a view from the width », *in Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 6232–6240, Curran Associates Inc., 2017.

[60] C. Wolf, « What is translation equivariance, and why do we use convolutions to get it ? », 2020. Billet de blog sur Medium.

[61] B. Jähne, éd., *Digital Image Processing*. Berlin, Heidelberg : Springer, 2005.

[62] X. Glorot et Y. Bengio, « Understanding the difficulty of training deep feedforward neural networks », *in Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, (Sardinia, Italy), p. 249–256, JMLR Workshop and Conference Proceedings, 2010.

[63] K. He, X. Zhang, S. Ren et J. Sun, « Delving Deep into Rectifiers : Surpassing Human-Level Performance on ImageNet Classification », *in International Conference on Computer Vision (ICCV)*, (Santiago, Chile), p. 1026–1034, IEEE, 2015.

[64] C. Olah, A. Mordvintsev et L. Schubert, « Feature Visualization », *Distill*, 2017.

[65] D. E. Rumelhart, G. E. Hinton et R. J. Williams, « Learning representations by back-propagating errors », *Nature*, vol. 323, no. 6088, p. 533–536, 1986.

[66] D. P. Kingma et J. Ba, « Adam : A Method for Stochastic Optimization », *in International Conference on Learning Representations (ICLR)*, (San Diego, California, USA), p. 1–15, 2015.

[67] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro et B. Recht, « The marginal value of adaptive gradient methods in machine learning », *in Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 4151–4161, Curran Associates Inc., 2017.

[68] S. Ioffe et C. Szegedy, « Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift », *in International Conference on Machine Learning (ICML)*, (Lille, France), p. 448–456, PMLR, 2015.

[69] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever et R. Salakhutdinov, « Dropout : A Simple Way to Prevent Neural Networks from Overfitting », *The Journal of Machine Learning Research*, vol. 15, no. 1, p. 1929–1958, 2014.

[70] Y. LeCun, « Generalization and network design strategies », *in Connectionism in perspective*, p. 1–20, R. Pfeifer and Z. Schreter and F. Fogelman and L. Steels, elsevier éd., 1989.

[71] H. Li, Z. Xu, G. Taylor, C. Studer et T. Goldstein, « Visualizing the Loss Landscape of Neural Nets », *in International Conference on Neural Information Processing Systems*, (Montreal, Canada), p. 1–11, 2018.

[72] J. Long, E. Shelhamer et T. Darrell, « Fully convolutional networks for semantic segmentation », *in Conference on Computer Vision and Pattern Recognition (CVPR)*, (Boston, MA, USA), p. 3431–3440, IEEE, 2015.

[73] A. Chaurasia et E. Culurciello, « LinkNet : Exploiting Encoder Representations for Efficient Semantic Segmentation », *IEEE Visual Communications and Image Processing (VCIP)*, p. 1–4, 2017.

[74] H. Zhao, J. Shi, X. Qi, X. Wang et J. Jia, « Pyramid Scene Parsing Network », *in Conference on Computer Vision and Pattern Recognition (CVPR)*, (Honolulu, HI, USA), p. 6230–6239, IEEE, 2017.

[75] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy et A. Yuille, « DeepLab : Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs », *Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, p. 834–848, 2018.

[76] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville et Y. Bengio, « Generative Adversarial Nets », *Advances in Neural Information Processing Systems 27*, p. 2672–2680, 2014.

[77] P. Wang et H. Huang, « Comparison analysis on present image-based crack detection methods in concrete structures », *in International Congress on Image and Signal Processing*, vol. 5, (Yantai, China), p. 2530–2533, IEEE, 2010.

[78] Y. Fujita, Y. Mitani et Y. Hamamoto, « A Method for Crack Detection on a Concrete Structure », *in International Conference on Pattern Recognition (ICPR)*, vol. 3, (Hong Kong, China), p. 901–904, IEEE, 2006.

[79] A. Ito, Y. Aoki et S. Hashimoto, « Accurate extraction and measurement of fine cracks from concrete block surface image », *in Conference of the Industrial Electronics Society (IECON)*, vol. 3, (Sevilla, Spain), p. 2202–2207, IEEE, 2002.

[80] H.-B. Yun, S. Mokhtari et L. Wu, « Crack Recognition and Segmentation Using Morphological Image-Processing Techniques for Flexible Pavements », *Transportation Research Record*, vol. 2523, no. 1, p. 115–124, 2015.

[81] J. Serra, *Image Analysis and Mathematical Morphology*. Londres : Academic Press, 1982.

[82] T. Yamaguchi, S. Nakamura et S. Hashimoto, « An efficient crack detection method using percolation-based image processing », *in Conference on Industrial Electronics and Applications*, (Singapore, Singapore), p. 1875–1880, IEEE, 2008.

[83] T. Yamaguchi et S. Hashimoto, « Fast crack detection method for large-size concrete surface images using percolation-based image processing », *Machine Vision and Applications*, vol. 21, p. 797–809, août 2010.

[84] A. R. Smith, « Tint fill », *in Proceedings of the 6th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '79, (New York, NY, USA), p. 276–283, Association for Computing Machinery, 1979.

[85] S.-N. Yu, J.-H. Jang et C.-S. Han, « Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel », *Automation in Construction*, vol. 16, no. 3, p. 255–261, 2007.

[86] J.-K. Oh, G. Jang, S. Oh, J. H. Lee, B.-J. Yi, Y. S. Moon, J. S. Lee et Y. Choi, « Bridge inspection robot system with machine vision », *Automation in Construction*, vol. 18, no. 7, p. 929–941, 2009.

[87] Y. Faula, *Extraction de caractéristiques sur des images acquises en contexte mobile : Application à la reconnaissance de défauts sur ouvrages dart*. Thèse de doctorat, INSA Lyon, Lyon, 2020.

[88] R. G. v. Gioi, J. Jakubowicz, J.-M. Morel et G. Randall, « LSD : a Line Segment Detector », *Image Processing On Line*, vol. 2, p. 35–55, 2012.

[89] R. Amhaz, S. Chambon, J. Idier et V. Baltazart, « Automatic Crack Detection on Two-Dimensional Pavement Images : An Algorithm Based on Minimal Path Selection », *Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, p. 2718–2729, 2016.

[90] B. E. Boser, I. M. Guyon et V. N. Vapnik, « A training algorithm for optimal margin classifiers », *in Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, (New York, NY, USA), p. 144–152, Association for Computing Machinery, 1992.

[91] N. Dalal et B. Triggs, « Histograms of oriented gradients for human detection », *in Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, (San Diego, CA, USA, USA), p. 886–893, IEEE, 2005.

[92] H. S. Munawar, A. W. A. Hammad, A. Haddad, C. A. P. Soares et S. T. Waller, « Image-Based Crack Detection Methods : A Review », *Infrastructures*, vol. 6, no. 8, 2021.

[93] K. Makantasis, E. Protopapadakis, A. Doulamis, N. Doulamis et C. Loupos, « Deep Convolutional Neural Networks for efficient vision based tunnel inspection », *in International Conference on Intelligent Computer Communication and Processing (ICCP)*, (Cluj-Napoca, Romania), p. 335–342, IEEE, 2015.

[94] K. Chaiyasarn, W. Khan, L. Ali, M. Sharma, D. Brackenbury et M. DeJong, « Crack Detection in Masonry Structures using Convolutional Neural Networks and Support Vector Machines », *in International Symposium on Automation and Robotics in Construction (ISARC)*, (Berlin, Germany), p. 118–125, 2018.

[95] D. Dais, I. E. Bal, E. Smyrou et V. Sarhosis, « Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning », *Automation in Construction*, vol. 125, p. 103606, 2021.

[96] M. Alipour, D. K. Harris et G. R. Miller, « Robust Pixel-Level Crack Detection Using Deep Fully Convolutional Neural Networks », *Journal of Computing in Civil Engineering*, vol. 33, no. 6, p. 04019040, 2019.

[97] V. Hoskere, Y. Narazaki, T. A. Hoang et B. F. Spencer Jr., « MaDnet : multi-task semantic segmentation of multiple types of structural materials and damage in images of civil infrastructure », *Journal of Civil Structural Health Monitoring*, vol. 10, no. 5, p. 757–773, 2020.

[98] Y. Ren, J. Huang, Z. Hong, W. Lu, J. Yin, L. Zou et X. Shen, « Image-based concrete crack detection in tunnels using deep fully convolutional networks », *Construction and Building Materials*, vol. 234, p. 117367, 2020.

[99] C. Zhang, C.-c. Chang et M. Jamshidi, « Simultaneous pixel-level concrete defect detection and grouping using a fully convolutional model », *Structural Health Monitoring*, p. 1475921720985437, 2021.

[100] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang et S. Wang, « DeepCrack : Learning Hierarchical Convolutional Features for Crack Detection », *Transactions on Image Processing*, vol. 28, no. 3, p. 1498–1512, 2019.

[101] Y. Liu, M.-M. Cheng, X. Hu, J.-W. Bian, L. Zhang, X. Bai et J. Tang, « Richer Convolutional Features for Edge Detection », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, p. 1939–1946, 2019.

[102] Q. Li, Q. Zou, J. Liao, Y. Yue et S. Wang, « Deep Learning with Spatial Constraint for Tunnel Crack Detection », *Computing in Civil Engineering 2019*, p. 393–400, 2019.

[103] Q. Mei, M. Gül et M. R. Azim, « Densely connected deep neural network considering connectivity of pixels for automatic crack detection », *Automation in Construction*, vol. 110, p. 103018, 2020.

[104] B. G. Pantoja-Rosero, D. Oner, M. Kozinski, R. Achanta, P. Fua, F. Perez-Cruz et K. Beyer, « TOPO-Loss for continuity-preserving crack detection using deep learning », *Construction and Building Materials*, vol. 344, p. 128264, 2022.

[105] X. Hu, L. Fuxin, D. Samaras et C. Chen, « Topology-Preserving Deep Image Segmentation », *in Conference on Neural Information Processing Systems (NeurIPS)*, (Vancouver, Canada), p. 1–11, 2019.

[106] P. Paleviius, M. Pal, M. Landauskas, U. Orinait, I. Timofejeva et M. Ragulskis, « Automatic Detection of Cracks on Concrete Surfaces in the Presence of Shadows », *Sensors*, vol. 22, no. 10, p. 3662, 2022.

[107] D. Loverdos et V. Sarhosis, « Automatic image-based brick segmentation and crack detection of masonry walls using machine learning », *Automation in Construction*, vol. 140, p. 104389, 2022.

[108] R. Santos, D. Ribeiro, P. Lopes, R. Cabral et R. Calçada, « Detection of exposed steel rebars based on deep-learning techniques and unmanned aerial vehicles », *Automation in Construction*, vol. 139, p. 104324, 2022.

[109] P. Savino et F. Tondolo, « Civil infrastructure defect assessment using pixel-wise segmentation based on deep learning », *Journal of Civil Structural Health Monitoring*, vol. 13, no. 1, p. 35–48, 2023.

[110] T. Kashiwa, K. Nagai, H. Tatsuta, H. Prendinger, K. Ibayashi et J. J. R. Guillamón, « Development of Delamination Detection System for Concrete Decks by Using Convolutional Neural Network », *International Conference on Experimental Mechanics (ICEM)*, vol. 2, no. 8, p. 418, 2018.

[111] J. J. Rubio, T. Kashiwa, T. Laiteerapong, W. Deng, K. Nagai, S. Escalera, K. Nakayama, Y. Matsuo et H. Prendinger, « Multi-class structural damage segmentation using fully convolutional networks », *Computers in Industry*, vol. 112, p. 103121, 2019.

[112] W. Deng, Y. Mou, T. Kashiwa, S. Escalera, K. Nagai, K. Nakayama, Y. Matsuo et H. Prendinger, « Vision based pixel-level bridge structural damage detection using a link ASPP network », *Automation in Construction*, vol. 110, p. 102973, 2020.

[113] B. Kim et S. Cho, « Automated Multiple Concrete Damage Detection Using Instance Segmentation Deep Learning Model », *Applied Sciences*, vol. 10, no. 22, p. 1–17, 2020.

[114] S. J. Schmugge, L. Rice, J. Lindberg, R. Grizziy, C. Joffey et M. C. Shin, « Crack Segmentation by Leveraging Multiple Frames of Varying Illumination », *in Winter Conference on Applications of Computer Vision (WACV)*, p. 1045–1053, IEEE, 2017.

[115] H. Bang, J. Min et H. Jeon, « Deep Learning-Based Concrete Surface Damage Monitoring Method Using Structured Lights and Depth Camera. », *Sensors*, vol. 21, no. 8, p. 1–15, 2021.

[116] S. Ren, K. He, R. Girshick et J. Sun, « Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks », *Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, p. 1137–1149, 2017.

[117] L. Yang, B. Li, W. Li, B. Jiang et J. Xiao, « Semantic Metric 3D Reconstruction for Concrete Inspection », *in Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, (Salt Lake City, UT, USA), p. 1624–16248, IEEE, 2018.

[118] T. Barisin, C. Jung, F. Müsebeck, C. Redenbach et K. Schladitz, « Methods for segmenting cracks in 3d images of concrete : A comparison based on semi-synthetic images », *Pattern Recognition*, vol. 129, p. 108747, 2022.

[119] O. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox et O. Ronneberger, « 3D U-Net : Learning Dense Volumetric Segmentation from Sparse Annotation », *in Medical Image Computing and Computer-Assisted Intervention MICCAI 2016* (S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal et W. Wells, éds), Lecture Notes in Computer Science, (Cham), p. 424–432, Springer International Publishing, 2016.

[120] S. Zhou et W. Song, « Deep learningbased roadway crack classification with heterogeneous image data fusion », *Structural Health Monitoring*, vol. 20, no. 3, p. 1274–1293, 2021.

[121] A. Zhang, K. C. P. Wang, Y. Fei, Y. Liu, S. Tao, C. Chen, J. Q. Li et B. Li, « Deep LearningBased Fully Automated Pavement Crack Detection on 3D Asphalt Surfaces with an Improved CrackNet », *Journal of Computing in Civil Engineering*, vol. 32, no. 5, p. 04018041, 2018.

[122] S. Zhou et W. Song, « Concrete roadway crack segmentation using encoder-decoder networks with range images », *Automation in Construction*, vol. 120, p. 103403, 2020.

[123] X. Chen, J. Li, S. Huang, H. Cui, P. Liu et Q. Sun, « An Automatic Concrete Crack-Detection Method Fusing Point Clouds and Images Based on Improved Otsus Algorithm », *Sensors*, vol. 21, no. 5, p. 1581, 2021.

[124] N. Otsu, « A Threshold Selection Method from Gray-Level Histograms », *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, p. 62–66, 1979.

[125] H. Liu, C. Yang, A. Li, S. Huang, X. Feng, Z. Ruan et Y. Ge, « Deep Domain Adaptation for Pavement Crack Detection », *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, p. 1669–1681, 2023.

[126] T. Durand, *Weakly Supervised Learning for Visual Recognition*. Thèse de doctorat, Pierre et Marie Curie, Paris, 2017.

[127] J. E. van Engelen et H. H. Hoos, « A survey on semi-supervised learning », *Machine Learning*, vol. 109, no. 2, p. 373–440, 2020.

[128] C. Feng, M.-Y. Liu, C.-C. Kao et T.-Y. Lee, « Deep Active Learning for Civil Infrastructure Defect Detection and Classification », *Computing in Civil Engineering*, p. 298–306, 2017.

[129] B. Settles, *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning, Cham : Springer International Publishing, 2012.

[130] S. Drouyer, « An 'All Terrain' Crack Detector Obtained by Deep Learning on Available Databases », *Image Processing On Line (IPOL)*, p. 1–18, 2019. https ://www.ipol.im/pub/art/2020/282/.

[131] M. J. Hallee, R. K. Napolitano, W. F. Reinhart et B. Glisic, « Crack Detection in Images of Masonry Using CNNs », *Sensors*, vol. 21, p. 4929, juil. 2021.

[132] R. Augustauskas et A. Lipnickas, « Improved Pixel-Level Pavement-Defect Segmentation Using a Deep Autoencoder », *Sensors*, vol. 20, no. 9, p. 2557, 2020.

[133] M. Long, Y. Cao, J. Wang et M. I. Jordan, « Learning transferable features with deep adaptation networks », *in International Conference on International Conference on Machine Learning (ICML)*, (Lille, France), p. 97–105, JMLR.org, 2015.

[134] Z. Dong, J. Wang, B. Cui, D. Wang et X. Wang, « Patch-based weakly supervised semantic segmentation network for crack detection », *Construction and Building Materials*, vol. 258, p. 120291, 2020.

[135] J. König, M. Jenkins, M. Mannion, P. Barrie et G. Morison, « A Weakly-Supervised Surface Crack Segmentation Method using Localisation with a Classifier and Thresholding », *arXiv :2109.00456 [cs]*, p. 1–7, 2021. arXiv : 2109.00456.

[136] Y. Inoue et H. Nagayoshi, « Crack Detection as a Weakly-Supervised Problem : Towards Achieving Less Annotation-Intensive Crack Detectors », *in International Conference on Pattern Recognition (ICPR)*, (Milan, Italie), p. 65–72, IEEE, 2021.

[137] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva et A. Torralba, « Learning Deep Features for Discriminative Localization », *in Conference on Computer Vision and Pattern Recognition (CVPR)*, (Las Vegas, NV, USA), p. 2921–2929, 2016.

[138] H. Maeda, T. Kashiyama, Y. Sekimoto, T. Seto et H. Omata, « Generative adversarial network for road damage detection », *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, p. 47–60, 2020.

[139] J.-Y. Zhu, T. Park, P. Isola et A. A. Efros, « Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks », *in International Conference on Computer Vision (ICCV)*, (Venice, Italy), p. 2242–2251, IEEE, 2017.

[140] W. Wang et C. Su, « Semi-supervised semantic segmentation network for surface crack detection », *Automation in Construction*, vol. 128, p. 103786, 2021.

[141] A. Tarvainen et H. Valpola, « Mean teachers are better role models : Weight-averaged consistency targets improve semi-supervised deep learning results », *in Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 1195–1204, Curran Associates Inc., 2017.

[142] R. Haralick, K. Shanmugam et I. Dinstein, « Textural Features for Image Classification », *Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, p. 610–621, 1973.

[143] Y. Xiao, J. Wu et J. Yuan, « mCENTRIST : A Multi-Channel Feature Generation Mechanism for Scene Categorization », *Transactions on Image Processing*, vol. 23, no. 2, p. 823–836, 2014.

[144] G. Decor, M. D. Bah, P. Foucher, P. Charbonnier et F. Heitz, « Détection d'anomalies dans des images de tunnels par forêts d'arbres aléatoires ou par apprentissage profond », *in Congrès Jeunes Chercheurs ORASIS*, (Saint-Dié-des-Vosges, France), p. 1–8, 2019.

[145] P. Foucher, G. Decor, F. Bock, P. Charbonnier et F. Heitz, « Evaluating of a Deep Learning Method for Detecting Exposed Bars From Images », *in Near Surface Geoscience Conference & Exhibition 2021 (NSG'21)*, (Bordeaux, France), p. 1–5, European Association of Geoscientists & Engineers, 2021. https ://doi.org/10.3997/2214-4609.202120191.

[146] M. Wang et W. Deng, « Deep visual domain adaptation : A survey », *Neurocomputing*, vol. 312, no. C, p. 135–153, 2018.

[147] G. Decor, P. Foucher, P. Charbonnier et F. Heitz, « IronNet : détection de fers apparents au sein de structures en béton armé », *in Congrès national sur la Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP)*, p. 1–3, 2020.

[148] U. Ozbulak, H. J. Lee, B. Boga, E. T. Anzaku, H.-m. Park, A. V. Messem, W. D. Neve et J. Vankerschaver, « Know Your Self-supervised Learning : A Survey on Image-based Generative and Discriminative Training », *Transactions on Machine Learning Research*, p. 1–45, 2023.

[149] G. Larsson, M. Maire et G. Shakhnarovich, « Colorization as a Proxy Task for Visual Understanding », *in Conference on Computer Vision and Pattern Recognition (CVPR)*, (Honolulu, HI), p. 840–849, IEEE, 2017.

[150] C. Doersch, A. Gupta et A. A. Efros, « Unsupervised Visual Representation Learning by Context Prediction », *in International Conference on Computer Vision (ICCV)*, (Santiago, Chile), p. 1422–1430, IEEE, 2015.

[151] K. He, H. Fan, Y. Wu, S. Xie et R. Girshick, « Momentum Contrast for Unsupervised Visual Representation Learning », *in Conference on Computer Vision and Pattern Recognition (CVPR)*, (Conférence Virtuelle), p. 9726–9735, IEEE, 2020.

[152] I. Misra et L. v. d. Maaten, « Self-Supervised Learning of Pretext-Invariant Representations », *in Conference on Computer Vision and Pattern Recognition (CVPR)*, (Conférence Virtuelle), p. 6706–6716, IEEE, 2020.

[153] A. Bardes, J. Ponce et Y. LeCun, « VICReg : Variance-Invariance-Covariance Regularization for Self-Supervised Learning », *in International Conference on Learning Representations (ICLR)*, (Conférence Virtuelle), p. 1–23, 2022.

[154] Y. Li, N. Wang, J. Shi, J. Liu et X. Hou, « Revisiting Batch Normalization For Practical Domain Adaptation », *in International Conference on Learning Representations (ICLR)*, (Toulon, France), p. 1–10, 2017.

[155] W.-G. Chang, T. You, S. Seo, S. Kwak et B. Han, « Domain-Specific Batch Normalization for Unsupervised Domain Adaptation », *in Conference on Computer Vision and Pattern Recognition (CVPR)*, (Long Beach, CA, USA), p. 7346–7354, IEEE, 2019.

[156] J. Frankle, D. J. Schwab et A. S. Morcos, « Training BatchNorm and Only BatchNorm : On the Expressive Power of Random Features in CNNs », *in International Conference on Learning Representations (ICLR)*, (Conférence Virtuelle), p. 1–28, 2021.

[157] X. Chen et A. Gupta, « Webly Supervised Learning of Convolutional Networks », *in International Conference on Computer Vision (ICCV)*, (Santiago, Chile), p. 1431–1439, IEEE, 2015.

[158] E. Bianchi et M. Hebdon, « Visual structural inspection datasets », *Automation in Construction*, vol. 139, p. 104299, 2022.

[159] M. Tual, P. Charbonnier et P. Foucher, « Robust B-spline surface estimation for tunnel lining modelling and equipment surveying », vol. 2021, p. 1–5, European Association of Geoscientists & Engineers, 2021.

[160] F. Barcet, M. Tual, P. Foucher et P. Charbonnier, « Using machine learning on depth maps and images for tunnel equipment surveying », *in Optical 3D Metrology (O3DM)*, vol. XLVIII-2/W2-2022, (Würzburg, Germany), p. 1–7, 2022.

[161] H. Kervadec, J. Bouchtiba, C. Desrosiers, E. Granger, J. Dolz et I. B. Ayed, « Boundary loss for highly unbalanced segmentation », *in International Conference on Medical Imaging with Deep Learning*, (London, England), p. 285–296, PMLR, 2019.

[162] E. Elizar, M. A. Zulkifley, R. Muharar, M. H. M. Zaman et S. M. Mustaza, « A Review on Multiscale-Deep-Learning Applications », *Sensors*, vol. 22, no. 19, p. 7384, 2022.

[163] L. Du, J. Tan, H. Yang, J. Feng, X. Xue, Q. Zheng, X. Ye et X. Zhang, « SSF-DAN : Separated Semantic Feature Based Domain Adaptation Network for Semantic Segmentation », *in International Conference on Computer Vision (ICCV)*, (Seoul, Korea (South)), p. 982–991, IEEE, 2019.

[164] P. Zhang, B. Zhang, T. Zhang, D. Chen, Y. Wang et F. Wen, « Prototypical Pseudo Label Denoising and Target Structure Learning for Domain Adaptive Semantic Segmentation », *in Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, (Conférence Virtuelle), p. 1–11, IEEE, 2021.

[165] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos et M. Valko, « Bootstrap your own latent, a new approach to self-supervised learning », *in Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, (Red Hook, NY, USA), p. 21271–21284, Curran Associates Inc., 2020.

[166] A. Ermolov, A. Siarohin, E. Sangineto et N. Sebe, « Whitening for Self-Supervised Representation Learning », *in International Conference on International Conference on Machine Learning (ICML)*, vol. 139, (Conférence Virtuelle), p. 3015–3024, 2021.

[167] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár et R. Girshick, « Segment Anything », 2023.

[168] D. Lin, Y. Cao, W. Zhu et Y. Li, « Few-Shot Defect Segmentation Leveraging Abundant Defect-Free Training Samples Through Normal Background Regularization And Crop-And-Paste Operation », *in International Conference on Multimedia and Expo (ICME)*, (Shenzhen, China), p. 1–6, IEEE, 2021.

[169] Y. Faula, S. Bres et V. Églin, « Détection et classification One-Class de défauts sur des surfaces bétonnées », *in Congrès national sur la Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP)*, (Conférence Virtuelle), p. 1–9, AFRIF, 2020.

[170] B. Frenay et M. Verleysen, « Classification in the Presence of Label Noise : A Survey », *Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, p. 845–869, 2014.

[171] Z. Zhang et M. R. Sabuncu, « Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels », *in International Conference on Neural Information Processing Systems*, (Montreal, Canada), p. 8792–8802, Curran Associates Inc., 2018.

Guillaume DECOR

Reconnaissance des formes pour l'inspection visuelle des tunnels

Résumé

Cette thèse vise à mettre en œuvre des méthodes de classification et de segmentation sémantique d'anomalies en tunnel à partir d'images photographiques ou de relevés laser. Pour ce faire, nous avons collecté et annoté des données provenant de plusieurs infrastructures, présentant des anomalies d'aspect et d'échelle différents. La première contribution de ce travail a été d'étudier un modèle de segmentation sémantique, avec une approche multi-échelle, entraîné puis évalué indépendamment sur chaque jeu de données. Ce modèle est capable de détecter l'essentiel des anomalies d'un tunnel dès lors que le biais de domaine entre les jeux d'apprentissage et de test est négligeable. Dans le cas contraire, nous avons mesuré des performances uniformément basses. Une seconde contribution a donc été de développer une méthode d'adaptation de domaine originale, reposant sur un ajustement de certains paramètres stratégiques d'un modèle entrainé sur un jeu source, et nécessitant d'annoter une petite quantité d'images du tunnel cible à analyser. Les performances obtenues par cette méthode approchent celles atteintes avec un modèle appris sur les données du tunnel évalué.

Mots-clés : reconnaissance des formes, inspection des tunnels, adaptation de domaine

Résumé en anglais

The aim of this thesis is to implement methods for the classification and semantic segmentation of tunnel anomalies based on photographic images or laser scans. To this end, we have collected and annotated data from several infrastructures, each presenting anomalies of different appearance and scale. The first contribution of this work was to study a multi-scale semantic segmentation model, trained and then evaluated independently on each dataset. This model is capable of detecting most of the anomalies in a tunnel as long as the domain shift between the training and test sets is negligible. Otherwise, we measured uniformly low performance. A second contribution was therefore to develop an original domain adaptation method, based on the adjustment of a few strategic parameters of a model trained on a source set, and requiring the annotation of a small number of images of the target tunnel to be analyzed. The performance achieved by this method approaches that of a model trained on the tunnel data being evaluated.

Keywords : pattern recognition, tunnel inspection, domain adaptation