

# Détection et classification One-Class de défauts sur des surfaces bétonnées

Yannick Faula

Stéphane Bres

Véronique Églin

Université de Lyon,  
INSA Lyon, LIRIS, UMR5205,  
F-69621, France

{yannick.faula, stephane.bres, veronique.eglin}@liris.cnrs.fr

## Résumé

*Aujourd'hui, certaines infrastructures font l'objet d'inspections régulières réalisées par acquisition d'images. Les experts ont besoin d'outils automatiques pour traiter ces images. La classification des défauts (fissures, éclatements, humidités, graffiti...) est rendue difficile par le manque de données représentatives. Nous proposons une nouvelle méthode basée sur la détection d'anomalies en utilisant les réseaux antagonistes génératifs conditionnels (GAN) et les motifs locaux binaires (LBP). La détection est suivie par une classification avec un système potentiellement évolutif. Les expérimentations montrent que notre détecteur obtient de meilleures performances que celles de l'état de l'art pour l'application à la détection de défauts.*

## Mots Clef

Réseaux antagonistes génératifs, classification indépendante One-Class, détection de défauts

## Abstract

*Nowadays, few infrastructures need regular structural health monitoring. Experts need automated methods based on image processing to analyze inspections by acquiring photographs. The classification of damages (crack, spall, humidity, graffiti...) is a big issue but good representative data is difficult to obtain. We propose a new method based on one-class anomaly detection. We use conditional generative adversarial networks (GAN) and local binary patterns (LBP). The detection is followed by a classification step with a potentially dynamic system. The experiments show our detector has better performance than state-of-art one-class detectors in the context of damage detection.*

## Keywords

Defaults detection, One-Class classification, GAN.

## 1 Introduction

L'utilisation de nouveaux vecteurs d'acquisition comme le drone permet une surveillance des ouvrages d'art à moindre coût, mais entraîne l'acquisition d'une énorme

quantité de photographies qu'il faut pouvoir traiter de manière automatique, efficace et rapide. La détection précoce des défauts dans les infrastructures permet d'éviter des dommages plus importants et des risques d'accident pour les constructions en béton qui sont initialement conçues pour être suffisamment durables pour supporter le poids, la pression et toutes les conditions météorologiques difficiles liées à leur utilisation. Les défauts (ou avaries) surfaciques représentent des défauts liés à une dégradation mécanique ou chimique. Tous ces défauts présentent des caractéristiques très différentes en texture, en forme et en couleur. Cela peut dépendre de plusieurs facteurs difficiles à identifier. Cependant, certains défauts sont assez répandus et font l'objet de sujet de recherche. Les types de défauts de surface les plus répandus sont les éclatements avec/sans fer apparent, l'humidité, la végétation. Le graffiti peut être également considéré comme défaut car il s'agit d'une dégradation visuelle de la surface. Elle peut avoir pour conséquence de réduire le contraste d'une fissure lorsque celle-ci était déjà présente au moment de l'application de la peinture aérosol. La surveillance d'infrastructures implique une détection de tous les défauts portant atteinte à la santé d'un ouvrage. Nous avons deux stratégies possibles : la détection à partir des caractéristiques de défauts connus ou la détection de l'anormalité à partir de surfaces sans défaut. La détection d'anomalies inclut l'anormalité, la nouveauté ou la détection d'aberrations. Dans chaque cas, le but est de détecter une situation différente de la normale. Comme domaine d'applications possibles, on peut mentionner la détection de comportement anormal en vidéo-surveillance ou la détection d'anomalies dans les images. Dans ce contexte, les instances représentant la "normalité" sont disponibles. Les autres instances, appartenant à une classe négative, ne sont pas présentes ou pas bien définies dans le jeu de données. Modéliser la normalité est une partie essentielle dans le processus de détection. Distinguer les situations dites "normales" des autres est donc un verrou scientifique important qui peut être envisagé sous l'angle d'une problématique de classification indépendante one-class.

L'utilisation d'une seule classe peut être imposée dès lors

qu'on ne dispose pas de suffisamment d'informations sur les autres situations. Dans notre domaine, les défauts représentent une très faible partie d'une infrastructure entière. La détection des défauts est utile et mène vers d'importantes décisions de maintenance. La difficulté dans la surveillance d'ouvrage d'art en béton est le nombre important de types de défauts existants. À cela s'ajoute le fait qu'un même type de défaut peut se présenter sous des formes très différentes.

Nous allons présenter les différentes méthodes existantes dans la littérature de classification de défauts des surfaces bétonnées puis, introduire notre solution basée sur les réseaux antagonistes génératifs (GAN) et les SVM one-class pour leur classification effective en y montrant les possibilités d'évolution dans le temps sans avoir recours à un ré-apprentissage complet des modèles.

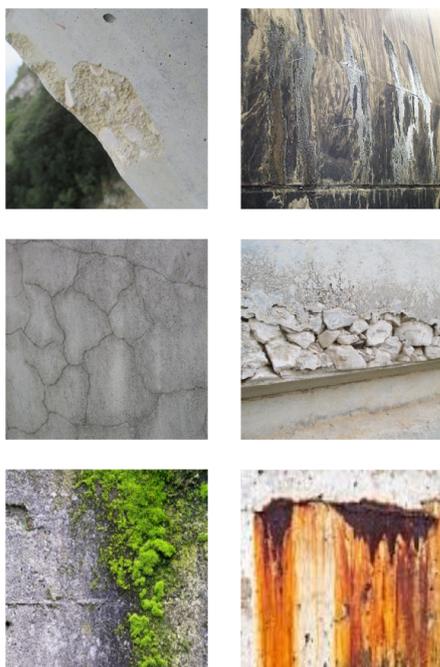


FIGURE 1 – Exemples de différent types de défauts

## 2 Détection de défauts

Compte tenu de la multiplicité de situations où les défauts des surfaces bétonnées sont observables, certaines ne sont pas traitées dans la littérature. Nous avons fait le constat que différentes techniques ont été utilisées dans l'état de l'art pour chaque nature de défaut. Nous présentons ici des algorithmes de détection dédiés aux défauts surfaciques exclusivement. Les fissures sont parfois détectées au même titre que les défauts de surface. Mais ici, nous ne considérons pas les fissures fines.

### 2.1 Etat de l'art des méthodes de détection de défauts surfaciques

La détection de défauts est une thématique de recherche de la vision par ordinateur qui a suscité un grand engouement

il y a une dizaine d'années. Dans [1] par exemple, les auteurs ont exploité des techniques classiques de traitement d'images comme les filtres gaussiens pour la détection de défauts surfaciques de type de bulles d'air. La déviation standard des valeurs en niveau de gris est utilisée pour détecter des traces d'humidités.

Les techniques de *machine learning* sont utilisées pour la détection de défauts en exploitant des descripteurs colorimétriques ou variationnels. Dans [2], les auteurs changent l'espace colorimétrique et testent trois algorithmes de classification (mixture de gaussiennes, réseaux de neurones artificiels et SVM) afin de détecter les défauts sur des surfaces de béton. Dans [3], la détection d'éclatements de béton est exécutée à l'aide du seuillage par entropie locale. L'estimation de la surface qu'occupe le défaut ainsi que la détection de fer apparent par analyse colorimétrique permet de compléter l'analyse.

À partir de 2015, la recherche s'est orientée vers de nouvelles techniques basées sur les réseaux de neurones à convolution (CNN). Ainsi, les auteurs dans [4] ont exploité un réseau convolutionnel à 2 couches de convolution et 3 couches complètement connectées de perceptron. En entrée du réseau, des caractéristiques variationnelles bas-niveau sont utilisées pour décrire chaque pixel de l'image. Le réseau sort ainsi une décision binaire de la présence d'un défaut pour chaque pixel. La détection est réalisée sur un patch de 5x5 autour des pixels. Plusieurs méthodes de requalification de CNN sont également proposées pour détecter et localiser des objets par propositions de régions. Différentes implémentations et améliorations ont été proposées dans la littérature ([5], [6], [7]). Dans [8], ils accomplissent une détection très spécifique comme les écrous corrodés et la délamination de fer. Dans le but de réduire le nombre d'exemples durant la phase d'apprentissage, les techniques de transfert learning avec le réseau VGG-16 sont exploitées pour la détection d'éclatements dans [9].

Peu de bases de données annotées portant sur la détection de défauts (en open data) sont rendues disponibles à la communauté scientifique. Les images sont le plus souvent privées comme le soulignent les auteurs dans [10].

Beaucoup de techniques dans la littérature sont spécialisées sur un type de défaut. Il est nécessaire de disposer de données annotées en quantité suffisante mais le constat récurrent est que celles-ci sont difficilement accessibles publiquement. C'est la raison pour laquelle, nous avons choisi de développer une approche de la détection de défauts utilisant exclusivement les exemples de défauts surfaciques et de surfaces sans défaut en ciblant ainsi des catégories très spécifiques. Notons enfin que dans le cas des défauts surfaciques, les classes de défauts ne sont généralement pas connues a priori. Il est nécessaire de se pencher vers la conception d'un système de reconnaissance dynamique ayant la capacité à s'auto-adapter aux nouvelles données.

L'état de l'art sur la détection de défaut montre qu'il existe peu de solutions qui tiennent compte à la fois de l'évolution de défauts selon la structure qui les contient, du nombre

d'échantillons disponibles, et de la qualité des sources. A contrario, alors que les classificateurs dits indépendants (comme les SVM one-class) semblent plus naturellement adaptés à ces types de problèmes, leur usage reste très modeste, sans doute à cause de la difficulté de leur paramétrisation et de leurs performances souvent moins élevées en classification que leurs homologues discriminants.

Ces observations nous ont conduits à privilégier une classification indépendante (One-Class SVM), permettant de traiter des données de catégories non équilibrées (unbalanced data), des données au départ inexistantes (classes inconnues rajoutées à la connaissance du système au fur et à mesure), des données évolutives (classes de défauts pouvant évoluer selon les acquisitions faites au fur et à mesure et acceptant une plus grande diversité de supports, de lieux, et de qualité d'images).

Cette approche présente également un intérêt important pour le ré-entraînement du système qui peut s'effectuer par parties seulement et de manière incrémentale : il n'est pas nécessaire de relancer un apprentissage complet pour le faire évoluer et améliorer la reconnaissance.

## 2.2 Apport de la classification one-class pour la détection de défauts

Les machines à vecteurs de support (SVM)[11] sont des algorithmes d'apprentissage supervisé utilisés pour la classification ou pour des tâches de régression. Un modèle SVM est entraîné afin de trouver une séparation linéaire optimale entre les classes. Cette séparation correspond à un hyperplan dans l'espace des descripteurs et doit maximiser la distance entre les classes. Les one-class SVM (OC-SVM) ont été popularisés par [12] mais apparaissent la première fois dans [13]. Pour les one-class SVM, une classe positive définit les données d'apprentissage et aucune donnée de classe négative n'est exploitée. Dans l'espace de représentation, une séparation avec un hyperplan doit être trouvée. On fait l'hypothèse que la distribution de la classe négative dans cette espace est une distribution autour du zéro. On revient alors au problème de départ (classification binaire) avec quelques modifications. Les OC-SVMs apparaissent dans beaucoup d'applications mais récemment, d'autres approches basées sur les réseaux de neurones sont proposées. Faire des hypothèses sur la classe négative est une tâche ardue. Dans [14], la classe négative est assimilée à une distribution normale centrée sur l'origine (comme originellement pour les OC-SVM). Leur réseau est un réseau classique à convolution. L'avant dernière couche simule la classe négative alors que la dernière couche est chargée de faire la différence entre la classe visée (classe positive) et la classe négative créée. Deux sorties donnent la probabilité pour un échantillon d'être dans une classe ou dans l'autre. Cette étape est ajoutée uniquement pendant la phase d'apprentissage.

Dans [15], un réseau antagoniste génératif (GAN) est utilisé pour la détection de nouveauté qui peut être suivie par une classification des échantillons inconnus. On peut as-

sembler cela à une méthode de classification hiérarchique. Rappelons que dans le cas des défauts surfaciques, les classes de défauts ne sont généralement pas connues a priori. Par conséquent, il est nécessaire de proposer une architecture dynamique ayant la capacité à s'auto-adapter aux nouvelles données. Le mot *dynamique* se réfère aux changements qui peuvent s'opérer au niveau de la conception des classes ou du classifieur.

Dans le cas de la classification des défauts, nous sommes face à un environnement stationnaire avec la possibilité d'ajouter de nouvelles classes. Les travaux de Khoi dans [16] ont montré la possibilité de faire évoluer un système de reconnaissance avec les one-class SVMs.

Les auteurs y ont proposé un système dynamique avec plusieurs OC-SVMs. La classification des différents objets est alors indépendante et permet plus de facilité pour faire évoluer les concepts.

## 3 Architecture proposée

L'architecture de reconnaissance de défauts surfaciques est basée sur un framework représenté à la figure 2. Notre architecture est similaire à une classification hiérarchique à deux niveaux. En plus d'utiliser les convolutions d'images au sein du GAN initial, elle exploite également les motifs binaires locaux (LBP)[17] ainsi qu'une série de classificateurs one-class SVM évolutifs.

Initialement, c'est un réseau de neurones de type conditionnel GAN (cGAN) qui est utilisé pour distinguer au sein des séries d'images acquises en mobilité celles qui contiennent des défauts de celles qui n'en contiennent pas. Les images prises par drone sont attendues avec une haute résolution. Par conséquent, nous avons divisé l'image en patches d'une taille 64x64 pixels. Le choix de la taille est motivé par le fait que certains défauts ne dépassent pas cette zone dans une image acquise dans les conditions prévues. En effet, la distance de la caméra à la surface est une contrainte forte qui motive notre choix de découpage. Trop près, la sécurité du drone ou de l'infrastructure n'est pas assurée et trop loin, la résolution de la prise de vue est trop faible pour détecter les différents défauts y compris les fissures fines.

### 3.1 Détection one-class

#### Réseaux antagonistes génératifs conditionnels (cGAN).

Les réseaux antagonistes génératifs sont introduits par [18] et sont une autre façon d'entraîner les modèles génératifs avec la back propagation. Les GANs sont composés de deux réseaux entraînés dans un processus antagoniste similaire à un jeu : un réseau générateur G et un réseau discriminatoire D, qui évalue la probabilité qu'un exemple a été généré avec G ou est un exemple réel. Comme introduit par [18], G reçoit en entrée un bruit aléatoire. Dans [19], les auteurs ont proposé de guider la génération afin de créer différentes classes identifiées en introduisant en entrée un vecteur connu et dépendant de sa sortie. Les auteurs réalisent des expérimentations en donnant en entrée les labels de la base de données. Plus tard, la même idée

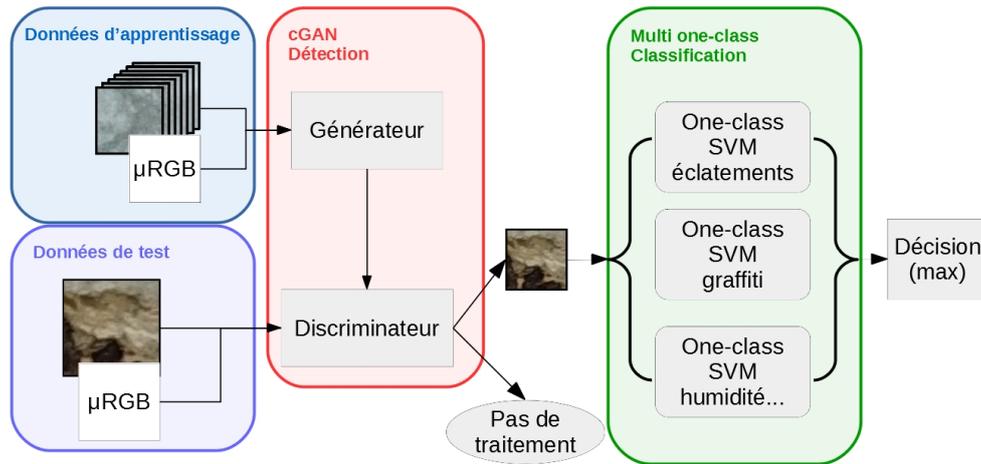


FIGURE 2 – Architecture hiérarchique pour la détection et la classification de défauts superficiels.

est reprise par [20] dans une implémentation permettant la translation d’images à images. Plusieurs avantages sont présentés dans leur architecture. En effet, le générateur est un réseau U-Net auto-encodeur avec des skip connections. Les réseaux générateur et discriminateur sont des réseaux totalement convolutifs. Dans notre processus, la première tâche est de séparer de manière efficace les patches avec défauts et les patches sans défaut. Plusieurs méthodes ont été testées et sont décrites rapidement dans la section 4. Le conditional GAN est la méthode qui donne les meilleurs résultats de séparation défaut/sans défaut (voir section 4). Notons que les fissures fines ne peuvent pas être détectées avec le GAN car leurs structures fines sont assimilées à un bruit quelconque dans les images. L’étape de détection de fissures est réalisée par notre algorithme FLASH dans [21]. Le générateur est exclusivement utilisé ici dans le but d’entraîner le discriminateur. Intuitivement, les entrées du générateur doivent avoir suffisamment d’information pour pouvoir reconstruire correctement une image. Le discriminateur pourra alors s’améliorer en créant les bonnes caractéristiques pour pouvoir faire la classification des images. Cependant, fournir trop d’informations au générateur n’est pas optimal car il sera capable de reconstruire des images complexes dans un processus d’apprentissage très rapide. En conséquence, le discriminateur ne sera pas capable de distinguer les images réelles des images générées car il fera face très vite à des images très similaires aux images réelles. La détection d’anomalies est réalisée uniquement par le discriminateur D.

**Le détecteur.** Nous suivons l’architecture proposée par [20]. Les patches étant plus petits, nous avons adapté les premières couches de convolution en diminuant le pas. Dans [20], les auteurs ont utilisé un patchGan d’une taille de 70x70 pixels qui permet au discriminateur de sortir une matrice de scalaires et non une valeur unique. La décision du discriminateur se fait donc sur un patch et non sur

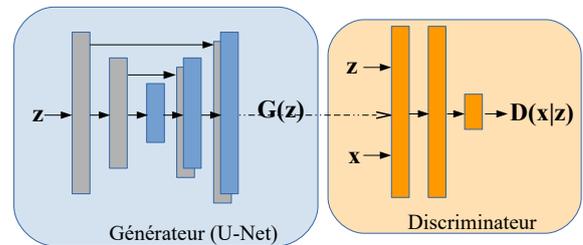


FIGURE 3 – Architecture du cGAN.

l’image complète. Cela permet d’obtenir des images avec moins de flou pour la génération des images. Ici, notre image est découpée en patches mais le discriminateur est entraîné comme un CNN classique et il produit une valeur unique pour chaque patch. Ainsi, pour traiter une image entière, elle est divisée en patches et une décision de détection est prise par chaque patch constituant l’image. Nos patches sont de taille 64x64 pixels. Au vue de la difficulté d’obtenir des images labellisées, les résultats sont présentés sur l’ensemble des patches et non sur les images entières. Le découpage de l’image peut être fait de différentes façons. Nous avons choisi de découper sans recouvrement afin de ne pas augmenter le temps de calcul. Nous n’avons pas étudié ici l’incidence du découpage, cela reste une perspective à envisager dans la poursuite des travaux. En utilisant uniquement le discriminateur D, les données d’origine  $x$  et l’information conditionnelle  $z$  montrées sur la figure 3, la détection est simple et est définie comme suit :

$$S(x|z) = \begin{cases} 1 \text{ (classe vise)} & \text{si } D(x|z) > T, \\ 0 \text{ (anormal)} & \text{sinon} \end{cases}, \quad (1)$$

où  $T$  est un seuil prédéfini.

L’information conditionnelle  $z$  est ici un ensemble de caractéristiques liés à la texture et/ou la couleur. Ces dernières sont décrites dans la partie suivante.

**Entrées conditionnelles  $z$ .** Les différents défauts diffèrent par leur couleur et/ou leur texture. Partant du principe que les LBP donnent une information de texture de l'image et en considérant uniquement les motifs considérés comme uniformes, nous disposons de 9 familles de motifs invariants en rotation. Sur la figure 4, nous pouvons visualiser la différence statistique entre une image sans défaut et avec défaut. L'analyse des familles de motifs montre que les motifs uniformes sont plus nombreux pour les images sans défaut.

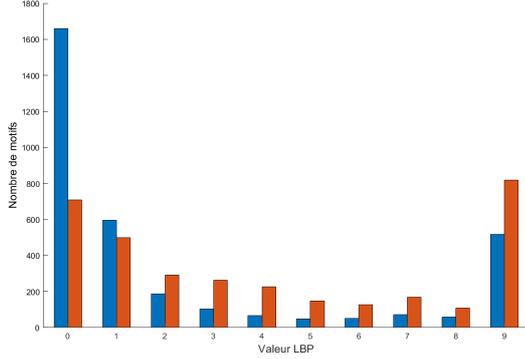


FIGURE 4 – Profil de l'opérateur LBP pour une image sans défaut (bleu) et avec défaut (rouge).

Les entrées de notre cGan sont au nombre de  $N$  couches : 3 pour une information liée à la colorimétrie et  $N - 3$  pour la texture créée par les points homogènes des opérateurs LBP. Une couche notée,  $C_{lbp}^i$ , est définie comme suit :

$$C_{lbp}^i = \begin{cases} 1 & \text{si } LBP_{P,R}^{riu2} = i, \\ 0 & \text{sinon} \end{cases} \quad (2)$$

On rappelle que l'opérateur LBP utilisé est fonction du nombre de voisins  $P$  autour d'un pixel central  $g_c$  et du rayon  $R$  de la zone locale à analyser, ainsi que d'un seuil  $K$ . L'opérateur s'exprime comme suit :

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & \text{si } U(LBP_{P,R}) \leq 2 \\ P + 1 & \text{sinon} \end{cases} \quad (3)$$

avec

$$s(x) = \begin{cases} 1, & x \geq K \\ 0, & x < K \end{cases} \quad (4)$$

avec  $U(LBP_{P,R})$  le nombre de transitions spatiales dans le motif. En conséquence,  $U(LBP_{P,R}) \leq 2$  représentent les motifs dit uniformes.

Toutes les entrées possibles, noté  $E$ , sont notées comme suit :

$$E = \left\{ \mu_R, \mu_G, \mu_B, C_{lbp}^0, \dots, C_{lbp}^9 \right\}$$

ou  $\mu_X$  est une couche pour laquelle toutes les valeurs sont normalisées à la moyenne du canal correspondant  $X$ . En pratique, pour un GAN classique  $z$  est un vecteur aléatoire. Ici, nous utilisons un GAN conditionnel (cGAN)

avec  $z = E$  une entrée connue sans information aléatoire intégrée. Dans notre cas,  $z$  est représenté par des matrices remplies avec une moyenne RGB entre -1 et 1 contenant une information colorimétrique, et/ou des matrices remplies avec des 0 et des 1 pour signaler la présence de points LBP contenant l'information de texture. Les différentes entrées du GAN sont discutées dans la section 4.

### 3.2 Classifieur multi one-class

Avec le manque d'exemples sur les défauts et le large panel des avaries que l'on peut rencontrer, nous avons choisi une architecture dynamique pour la classification. Nous avons choisi l'architecture proposée par [16]. La figure 2 décrit cette architecture adaptée à notre application. Plusieurs OC-SVMs avec les mêmes descripteurs sont utilisés pour classifier les différentes catégories de défauts. Les OC-SVMs sont indépendants et leur apprentissage peut être réalisé à tout moment dès lors qu'un expert intervient. On peut alors faire évoluer continuellement le système et bénéficier de la capacité incrémentale des SVMs.

#### Calcul des descripteurs pour le multi one-class.

Comme pour le fonctionnement des GANs, les descripteurs choisis pour les classifieurs One-Class SVM sont également basés sur une information colorimétrique et de texture. Afin de réduire les temps de calcul moyens, nous avons choisi de réutiliser les valeurs déjà calculés par l'opérateur LBP. Les moyennes des canaux  $R$ ,  $V$  et  $B$  sont ensuite incluses comme descripteurs pour la couleur. Au total nous avons exploité 13 descripteurs.

### 3.3 Prédictions de la classification

Le modèle proposé par [16] prédit les classes en sélectionnant la valeur maximale en sortie de classification pour chacun des OC-SVM. Cependant, comme les classifieurs sont indépendants, le seuil de décision peut être différent même si les mêmes descripteurs sont utilisés. C'est pourquoi, nous avons décidé de transformer la valeur prédite  $f_i$  par le  $i^{me}$  one-class, par sa version normalisée et centrée  $\hat{f}_i$ . Suivant une distribution normale standard cette valeur peut se définir par l'équation suivante :

$$\hat{f}_i = \left( \frac{f_i - \mu_i}{\sigma_i} \right)^2 \quad (5)$$

$\mu_i$  et  $\sigma_i$  sont respectivement la moyenne et la déviation standard calculées avec le jeu de données d'apprentissage utilisé pour le  $i^{me}$  one-class SVM. Ensuite, nous pouvons comparer chaque valeur, la valeur la plus proche de 0 étant celle qui représente le plus la classe à assigner.

De plus, nous pouvons détecter de nouvelles classes en rejetant tout échantillon ayant une valeur très haute. Un seuil peut être déterminé avec les données d'apprentissage.

## 4 Expérimentations

### 4.1 Scénarios pour la détection des défauts

Nous avons sélectionné trois scénarios lors de la phase de test :

Combinaison des entrées	S1	S2	S3
$\mu RGB$	✓	✓	
$\mu RGB + C_{lbp}^0 + \dots + C_{lbp}^9$	✓	✓	✓
$\mu RGB + C_{lbp}^0 + C_{lbp}^1$	✓	✓	
$C_{lbp}^0 + \dots + C_{lbp}^9$			✓
googlenet			✓

TABLE 1 – Descripteurs utilisés pour la détection. L’entrée de S1 et S2 est  $z$ . Pour S3, il s’agit des entrées des OC-SVMs. Rappelons que les entrées sont sous forme d’histogrammes et non par couches pour S3.

1. **S1.** Le discriminateur est uniquement utilisé avec différentes entrées conditionnelles  $z$  et l’image à tester  $x$  suivant les notations de la figure 3. Ce scénario est représenté par les courbes pleines dans la figure 5.
2. **S2.** Le discriminateur est utilisé comme dans le premier scénario avec les mêmes entrées conditionnelles  $z$ . Au lieu de donner l’image  $x$ , nous introduisons, comme dans la stratégie adoptée par [15], la sortie du générateur  $G(z)$ . L’idée est que la génération de l’image avec défaut échoue et le discriminateur est capable de mieux séparer l’ensemble des images. Ce scénario est représenté par les courbes en tirets dans la figure 5.
3. **S3.** Le dernier scénario est une comparaison avec des one-class SVM classiques avec différents descripteurs. Ces derniers sont globalement les mêmes que ceux utilisés dans les deux premiers scénarios. L’implémentation diffère par l’injection de l’information. En effet, on utilise un histogramme en entrée des OC-SVMs pour l’information de texture comme présenté sur la figure 4. Nous avons ajouté à cela des descripteurs issus du CNN GoogleNet [22]. Ce scénario est représenté par les courbes en pointillés dans la figure 5.

Pour chaque scénario, nous décrivons dans le tableau 1 les différents descripteurs utilisés.

## 4.2 Implémentation

Les deux parties du système ont été testées séparément avec les algorithmes de l’état de l’art. La première partie est implémentée avec le framework *Chainer* et les calculs de LBP sont réalisés à l’aide de la librairie *OpenCV*. Le conditional GAN est entraîné *from scratch* avec les mêmes paramètres utilisés dans [20]. Les fonctions *loss* utilisées sont également les mêmes décrites dans [20]. En reprenant les notations de la section précédente, la fonction objective s’écrit :

$$G^* = \arg \min_G \max_D \mathbb{E}_{x,z} [\log(D(x|z))] + \mathbb{E}_z [\log(1 - D(G(z)|z))] + \lambda \mathbb{E}_{x,z} [\|x - G(z)\|_1]$$

où le terme avec la distance L1 permet au générateur d’être plus proche de la vérité terrain. Le générateur  $G$  essaie de minimiser cette fonction alors que le discriminateur  $D$  tente de la maximiser. Concernant la deuxième partie, les hyperparamètres pour les OC-SVMs sont obtenus par optimisation selon l’algorithme *Grid Search*. Le noyau pour les OC-SVMs est gaussien radial. Les tests à destination de l’optimisation peuvent être faits en utilisant des données nouvelles et labellisées de la classe du OC-SVM considéré, ou en utilisant les données d’apprentissage des autres classes. On cherche alors à avoir le meilleur rejet possibles sur tous les échantillons disponibles. En effet, l’optimisation est difficile pour un OC-SVM parce que plusieurs séparations sont possibles dans l’espace des descripteurs.

Concernant l’opérateur LBP, le seuil  $K$  est fixé à 5 avec  $P = 8$  et  $R = 3$ . Des tests ont été réalisés avec un seuil  $K = 15$ . Nous avons comparé la première partie du système avec un OC-SVM construit avec différents descripteurs. Ces derniers sont parfois issus de réseau de neurones comme VGG-16 ou GoogleNet [22]. Nous avons gardé la dernière couche de GoogleNet avant la couche de décision comme vecteur descripteur (vecteur de 256 valeurs pour nos patches) puis injecté dans le OC-SVM. Les poids utilisés par GoogleNet sont ceux du réseau entraîné sur la base de données ImageNet.

## 4.3 Base de données et métriques

Notre base de données est composée d’images de surface de béton. Ces dernières sont divisées principalement en 3 classes : normal, éclatement et graffitis. On retrouve des exemples d’éclatements sur la figure 1. Nous utilisons également des images considérées comme normales mais présentant des ombres afin de tester la robustesse du système. Au total, plus de 30 grandes images sont utilisées permettant d’extraire plus de 16000 patches. Nous avons divisé le jeu de données d’images dites normales en utilisant approximativement 70% des patches pour l’apprentissage et le reste pour la phase de test. De la même manière, chaque classe de défaut est divisée en 2 parties. Le tableau 2 montre le nombre de patches utilisés pour les expériences. Une cross-validation a été effectuée en réalisant 5 exécutions différentes avec 5 tirages aléatoires comprenant les mêmes proportions de patches décrites précédemment.

TABLE 2 – Répartition de notre base de données

Classe	Normal	Défaut	
		Éclatement	Graffiti
<b>Apprentissage</b>	12039	500	500
<b>Total</b>	16271	1393	2751

Les métriques utilisées sont l’aire sous la courbe (AUC) et les courbes ROC présentent le taux de bons négatifs (spécificité) par rapport au taux de bons positifs.

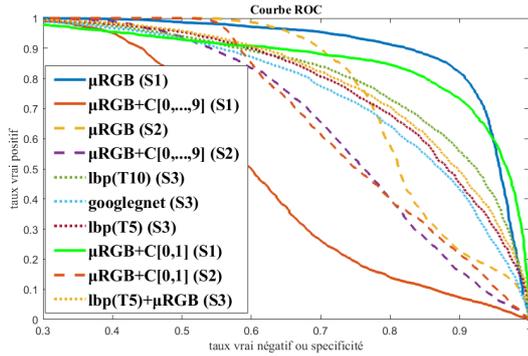


FIGURE 5 – Courbe ROC pour la détection d'anomalies en utilisant notre discriminateur et les OC-SVMs selon les scénarios (S1, S2 et S3).

#### 4.4 Résultats

**Résultats de la détection de défauts.** La figure 5 nous montre les différentes courbes obtenues dans des configurations différentes. Les OC-SVMs sont comparés avec différents descripteurs et la distance à l'hyperplan sert de seuil de décision pour construire la courbe. Différentes stratégies avec le cGAN ont été adoptées. Le profil LBP suggère de prendre en considération certains types de valeur LBP. Nous avons considéré les sorties du générateur comme dans [15]. Sur les figures 5 et 7, l'extension "(Sn)" indique le scénario utilisé avec le descripteur associé. Contrairement à [15], la version incluant le générateur en entrée du discriminateur ne donne pas les meilleurs résultats. Le générateur déforme très peu les échantillons appartenant à la "classe anormale" car les images restent assez homogènes dans l'ensemble. Il aurait plutôt pour effet de flouter la version "anormale". Donner trop d'informations au cGAN n'est pas une bonne stratégie. Les échantillons négatifs (n'appartenant pas à la classe visée) ont dans notre cas des textures assez différentes. C'est pourquoi, les OC-SVMs avec des descripteurs LBP sont assez performants. Cependant, si l'on considère la couleur, il est plus difficile de différencier un graffiti d'un mur normal. En d'autres termes, ajouter une information colorimétrique n'est pas suffisant pour classifier tous les types de défauts.

Dans l'ensemble, les meilleurs résultats sont obtenus avec la moyenne RGB. Cependant, dans la figure 7, les tests effectués avec quelques images d'ombres montrent qu'ajouter une information indépendante de l'illumination (comme les descripteurs LBP), améliore la robustesse significativement. Les couches de LBP utilisées dans ce cas sont  $C_{lbp}[0]$ ,  $C_{lbp}[1]$ .

Pour conclure, les OC-SVMs sont globalement moins performants que le cGAN sur la même base d'apprentissage pour l'ensemble des descripteurs utilisés. Le changement des descripteurs en entrée ne permet pas une nette amélioration des résultats. Cependant, les OC-SVMs restent plus performants que le cGAN sur la combinaison de tous les descripteurs (moyenne RGB et toutes les familles de LBP).

En réalité, les deux systèmes n'utilisent pas l'information de la même manière et par conséquent, le choix des descripteurs est plutôt dépendant de la stratégie adoptée. En effet, ajouter des descripteurs pertinents aux OC-SVMs lui permettra de s'améliorer. Ajouter ces mêmes descripteurs au cGAN peut inversement diminuer ses performances. La qualité du discriminateur (et donc du cGAN) va dépendre de la capacité du générateur à fournir des images réalistes, et ce, tout le long de l'apprentissage. Des descripteurs permettant une génération proche d'une image réaliste ne vont pas améliorer le discriminateur, et celui-ci n'améliorera pas sa capacité à discerner un patch faux d'un vrai. A contrario, une combinaison plus simple de descripteurs, c'est-à-dire avec un peu moins d'informations, permettra de rendre le discriminateur plus performant car son apprentissage sera progressif (comme celui du générateur) et plus robuste. Par conséquent, le choix des descripteurs dépend de la stratégie adoptée. Le cGAN peut être utilisé pour la détection d'anomalies (au sens large), pour améliorer les résultats sous conditions d'avoir des descripteurs adaptés. Dans nos expérimentations, il obtient, toutes combinaisons confondues, les meilleurs résultats. À l'inverse, un mauvais choix de descripteurs peut entraîner une diminution nette des résultats. Si le choix des descripteurs est très délicat, une solution "pratique" serait de se tourner vers les OC-SVMs plus stables à l'ajout de descripteurs mixtes (texture et couleur).

Nous avons comparé notre méthode avec le réseau décrit dans [14] en terme d'accuracy. Ce réseau a été entraîné avec les mêmes données présentées à notre cGAN. Nous obtenons une meilleure accuracy (rapport entre le nombre de patches bien classés et le nombre total des patches) avec 0.82 alors que le modèle de [14] a obtenu 0.69. La modélisation que les auteurs font de la classe négative peut être complexifiée pour notre application. Des exemples de détection sur une image complète sont présentés sur la figure 6.

TABLE 3 – Résultats de la classification.

Methodes	Accuracy (2 classes)
Multi class	.94
Multi One-Class	.91

**Classification.** Notre phase de classification est confrontée à un SVM multi-classe standard. Les hyperparamètres sont aussi optimisés avec l'algorithme Grid Search et nous avons gardé les mêmes descripteurs pour les 2 systèmes. Le tableau 3 montre que le système multi-classe est meilleur sans surprise. Dans le cas d'une classification binaire, le multi-classe SVM peut séparer efficacement les classes car le coût de la séparation est calculé en fonction des deux classes d'échantillons. Mais avec plus de données et plus de classes, la performance décroît généralement.

Le système dynamique a besoin de plus de recherche à cause du manque de données. Mais il est prêt à recevoir de nouvelles données. De plus, le système permet de s'adap-

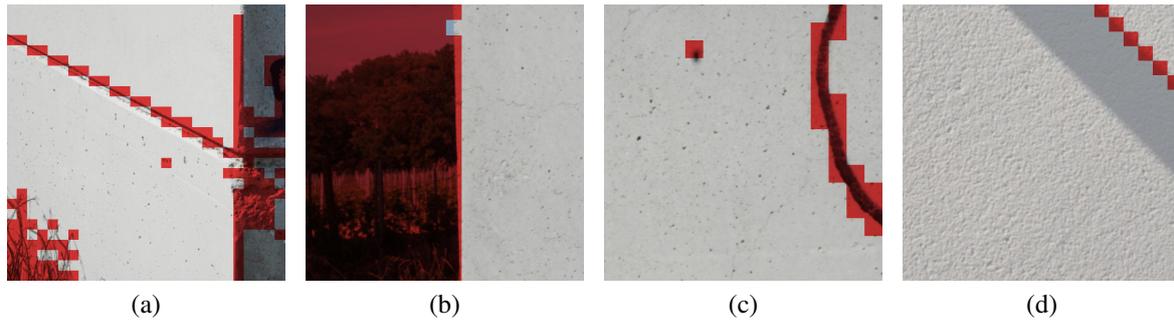


FIGURE 6 – Exemples de détection de défauts par cGAN ( $z=\mu RGB$ ) sur des images entières (a) Bonne détection de végétation, graffiti, éclatement et élément de structure (b) rejet du fond (c) Détection de graffiti (d) Problème d’ombre

ter à plusieurs types d’infrastructures comme les infrastructures métalliques.

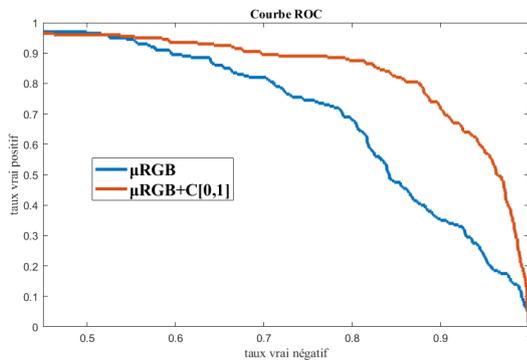


FIGURE 7 – Robustesse aux zones d’ombres

## 5 Conclusion

Bien que les fissures soient le défaut le plus important, tous les autres types de défauts ne sont pas traités dans la littérature actuelle. Une multitude de systèmes existe pour des défauts spécifiques comme l’éclatement de béton. L’utilisation de réseaux de neurones reste le meilleur moyen d’obtenir de bons résultats pour de gros défauts. Cependant, les petits défauts surfaciques sont difficiles à détecter pour un réseau de neurones. C’est pourquoi, nous proposons également des techniques plus classiques liées à l’analyse de texture. Notre système repose sur une analyse par patch qui permet de détecter de petits défauts. Il repose sur une approche multi one-class SVM et sur l’utilisation de la partie discriminante d’un GAN. Les résultats sont très intéressants car ils mettent en évidence la capacité d’un cGAN à améliorer la détection de l’anormalité à l’aide de descripteurs (ici les LBP). Plus précisément dans notre application, notre système permet de détecter tous les types de défauts et mènent vers une classification des défauts où le multi OC-SVM montre de bons résultats. Les recherches sur le choix des descripteurs, des hyperparamètres, ou la transformation des valeurs prédites peuvent amener à de

meilleurs résultats.

## Références

- [1] Z. Zhu and I. Brilakis, “Defects Detection & Assessment in Concrete Surfaces,” in *ResearchGate*, Jan. 2008, pp. 441–450.
- [2] Son Hyojoo, Kim Changmin, and Kim Changwan, “Automated Color Model-Based Concrete Detection in Construction-Site Images by Using Machine Learning Algorithms,” *Journal of Computing in Civil Engineering*, vol. 26, no. 3, pp. 421–433, May 2012.
- [3] Stephanie German, Ioannis Brilakis, and Reginald DesRoches, “Rapid entropy-based detection and properties measurement of concrete spalling with machine vision for post-earthquake safety assessments,” *Advanced Engineering Informatics*, vol. 26, no. 4, pp. 846–858, Oct. 2012.
- [4] K. Makantasis, E. Protopapadakis, A. Doulamis, N. Doulamis, and C. Loupos, “Deep Convolutional Neural Networks for efficient vision based tunnel inspection,” in *2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, Sept. 2015, pp. 335–342.
- [5] Ross Girshick, “Fast R-CNN,” *arXiv :1504.08083 [cs]*, Sept. 2015, arXiv : 1504.08083.
- [6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, “Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks,” *arXiv :1506.01497 [cs]*, Jan. 2016, arXiv : 1506.01497.
- [7] Joseph Redmon and Ali Farhadi, “YOLO9000 : Better, Faster, Stronger,” *arXiv :1612.08242 [cs]*, Dec. 2016, arXiv : 1612.08242.
- [8] Young-Jin Cha, Wooram Choi, Gahyun Suh, Sadeh Mahmoudkhani, and Oral Büyüköztürk, “Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 731–747, Nov. 2017.

- [9] Yuqing Gao and Khalid M. Mosalam, "Deep Transfer Learning for Image-Based Structural Damage Recognition," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 748–768, Sept. 2018.
- [10] Christian Koch, Kristina Georgieva, Varun Kasireddy, Burcu Akinci, and Paul Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 196–210, Apr. 2015.
- [11] Vladimir Vapnik, "Statistical Learning Theory | Statistics Special Topics | General & Introductory Statistics | Subjects | Wiley," 1998.
- [12] Yunqiang Chen, Xiang Sean Zhou, and T. S. Huang, "One-class SVM for learning in image retrieval," in *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, Oct. 2001, vol. 1, pp. 34–37 vol.1.
- [13] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, July 2001.
- [14] Poojan Oza and Vishal M. Patel, "One-Class Convolutional Neural Network," *IEEE Signal Process. Lett.*, vol. 26, no. 2, pp. 277–281, Feb. 2019, arXiv : 1901.08688.
- [15] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli, "Adversarially Learned One-Class Classifier for Novelty Detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [16] Anh Khoi Ngo Ho, Véronique Eglin, Nicolas Rago, and Jean-Yves Ramel, "A multi-one-class dynamic classifier for adaptive digitization of document streams," *International Journal on Document Analysis and Recognition*, pp. 1–18, May 2017.
- [17] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, July 2002.
- [18] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative Adversarial Nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, Cambridge, MA, USA, 2014, NIPS'14, pp. 2672–2680, MIT Press, event-place : Montreal, Canada.
- [19] Mehdi Mirza and Simon Osindero, "Conditional Generative Adversarial Nets," *arXiv :1411.1784 [cs, stat]*, Nov. 2014, arXiv : 1411.1784.
- [20] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 5967–5976.
- [21] Yannick Faula, Stéphane Bres, and Véronique Eglin, "FLASH : A New Key Structure Extraction used for Line or Crack Detection :," in *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Funchal, Madeira, Portugal, 2018, pp. 446–452, SCITEPRESS - Science and Technology Publications.
- [22] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1–9, ISSN : 1063-6919, 1063-6919.