Corruption synthétique d'image pour la détection d'anomalies à l'aide d'auto-encodeurs convolutifs

Anne-Sophie Collin Christophe De Vleeschouwer

Institut ICTEAM, UCLouvain

{anne-sophie.collin, christophe.devleeschouwer}@uclouvain.be

Résumé

De manière à détecter des anomalies, un auto-encodeur peut être entraîné à réaliser une projection depuis une image arbitraire, c.-à-d. avec ou sans défaut, vers une image saine, c.-à-d. sans défaut. Le résidu de reconstruction quantifie alors la vraisemblance qu'une région soit anormale. Dans un contexte où seules des images saines sont disponibles pour l'entraînement, nous proposons de corrompre ces dernières à l'aide de bruit synthétique. Nos expériences démontrent qu'un bruit en forme de tache indépendant du contenu de l'image, combiné à l'ajout de skip-connections à l'auto-encodeur, généralise sur les défauts réels, améliorant de ce fait la détection d'anomalies.

Mots Clef

Détection d'anomalies, non supervisé, auto-encodeur.

Abstract

In order to detect anomalies, an autoencoder can be trained to perform an image-to-image mapping from an arbitrary image, i.e. with or without defect, towards a clean image, i.e. without defect. Then, the residual map of reconstruction can be interpreted as the likelihood that a region is abnormal. In a scenario where only clean images are available for training, we suggest to corrupt them with synthetic noise. We show that a structured stain-shaped noise, independent of the image content, combined with the addition of skip-connections to the autoencoder, generalizes to real defects and leads to better anomaly detection.

Keywords

Anomaly detection, unsupervised, autoencoder.

1 Introduction

À première vue, la détection d'anomalies est une tâche qui semble parfaitement convenir à un schéma de classification binaire, distinguant les images saines de celles comportant des défauts. En outre, les différents types de défauts peuvent également mener à une classification plus raffinée en de multiples catégories. Cependant, la création d'une base de données nécessaire à la supervision d'un tel classificateur constitue une charge de travail considérable de par la définition intrinsèque du problème : les images comportant un défaut sont difficiles à collecter à cause de leur rareté. De plus, l'apparence d'un défaut peut être très variable ce qui rend difficile un échantillonage suffisamment représentatif de ce type de défaut.
Pour ces différentes raisons, il est pertinent d'envisager la détection d'anomalies dans un contexte d'apprentissage

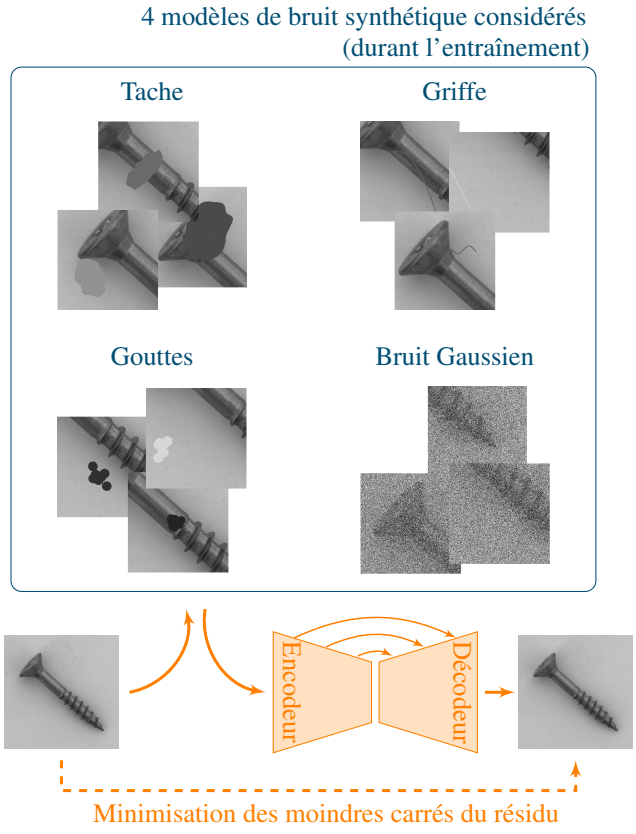

FIGURE 1 – La corruption des images durant l'entraînement (via le modèle Tache, Griffe, Gouttes ou Bruit Gaussien) ainsi que l'ajout de *skip-connections* permettent de diminuer l'erreur de reconstruction des structures saines d'une image tout en évitant la convergence du modèle vers l'opérateur identité (comme discuté dans la section 4.2).

non supervisé. Une solution non supervisée pourra être utilisée en tant que telle, ou pour faciliter la constitution d'une base de donnée destinée à une approche supervisée.

Concrètement, notre travail s'attache à fournir un outil capable de reconstruire une image saine, sans défaut ou anomalie, aussi similaire que possible d'une image arbitraire, avec ou sans défaut. À cette fin, un auto-encodeur convolutif est entraîné à reconstruire des images appartenant exclusivement à la distribution des images considérées comme normales, et le résidu de reconstruction est utilisé pour localiser les anomalies.
Traditionellemment, de par son goulot d'étranglement, l'auto-encodeur est amené à apprendre des représentations internes compressées des données d'entrainement. Cette réduction de dimension tend à régulariser la reconstruction des images de sortie par rapport aux structures normales. Afin d'obtenir une reconstruction plus détaillée, nous proposons d'introduire des *skip-connections* entre les couches convolutives de même dimension. Cette pratique facilite la propagation des structures fines de l'image au travers de l'auto-encodeur sans passer via la réduction de dimension [1].
Néanmoins, sans précaution, ces *skip-connections* peuvent réduire l'auto-encodeur à un opérateur identité en court-circuitant le passage de l'information par la réduction de dimension (phénomène discuté dans la section 4.2). En propageant à la fois les structures normales et anormales de l'image par les skip-connections, les régions anormales d'une image seraient reconstruites à l'identique rendant ainsi leur détection impossible par le résidu de la reconstruction. Pour éviter ce problème, nous proposons d'entraîner le réseau à reconstruire une image saine à partir d'une version corrompue de cette dernière.

En termes de contributions, notre travail discute le bénéfice découlant de l'ajout de *skip-connections* combiné à la corruption des images saines durant l'entrainement de l'auto-encodeur. Nous étudions cette formulation à deux égards. D'une part, nous comparons plusieurs types de bruit synthétique et évaluons leur impact sur la reconstruction ainsi que sur la détection d'anomalies en tant que telle. D'autre part, une fois le modèle de reconstruction établi, plusieurs stratégies sont considérées pour l'exploiter afin de décider de la présence d'anomalies. Cette opération peut s'effectuer soit au niveau de l'image entière, soit au niveau du pixel.

Dans la suite du papier, la Section 2 positionne notre travail par rapport à l'état de l'art. La Section 3 introduit les architectures d'auto-encodeurs étudiées ainsi que les modèles de corruption synthétiques utilisés. Enfin, la Section 4 présente et discute les résultats expérimentaux obtenus.

2 Travaux connexes

Comme présenté dans les travaux de Pimentel et al. [2] et de Chandola et al. [3], la détection d'anomalies est un problème bien connu dont les domaines d'application sont variés. Les approches envisagées pour le résoudre sont multiples mais nous avons décidé de nous focaliser sur les méthodes procédant par reconstruction régularisée (c.-à-d. dans l'espace des images sans défaut) de l'image analysée. En comparaison avec des méthodes pour lesquelles la décision d'appartenance à la classe saine est effectuée dans un espace autre que celui de l'image [4, 5, 6], l'approche par reconstruction offre l'avantage de pouvoir identifier les pixels ayant contribué au rejet de l'image de la classe saine.

Dans les approches par reconstruction, le réseau utilisé afin d'effectuer la reconstruction est généralement un auto-encodeur dont la tâche est double. D'une part, l'encodeur sert à projeter une image arbitraire vers un espace latent de plus faible dimension. Et d'autre part, le décodeur est entraîné à reconstruire l'image d'entrée hors de cette représentation compressée. Lors de l'entraînement, la minimisation de l'erreur de reconstruction des images saines permet de contraindre la représentation interne à être spécifique à l'espace des images sans défaut.
La fonction de coût la plus employée est l'erreur de reconstruction par les moindres carrés des résidus. Cependant, il est bien connu que les images qui résultent de cet entrainement sont généralement floues. Pour palier à ce problème, Bergmann et al. [7] proposent d'utiliser une fonction de coût basée sur la similarité structurelle (SSIM).

Des alternatives ont proposé d'utiliser des réseaux génératifs antagonistes (GANs) afin d'échantillonner la distribution des images saines [8, 9, 10, 11, 12]. De par l'implication d'un discriminateur dans la fonction de coût, les images produites par le réseau génératif comportent des structures plus nettes que pour un auto-encodeur minimisant la reconstruction au sens de la somme des moindre carrés du résidu. Cependant, les GANs sont particulièrement complexes à entraîner à cause de leur forte tendance à converger vers une reconstruction unique (connue en anglais sous le terme de *mode collapse*). De plus, la difficulté d'exclure les défauts du modèle génératif dégrade les performances de la méthode [12] .

Traditionnellement, il est attendu de l'auto-encodeur que la réduction de dimension à elle seule restreigne la reconstruction à l'espace d'images saine. En pratique cependant, une partie de la difficulté réside dans le fait que l'auto-encodeur n'est pas contraint explicitement à ne pas reproduire des contenus anormaux. Il est ainsi difficile de restreindre le domaine d'arrivée uniquement aux images saines. En d'autres mots, l'absence de structures caractérisant un défaut durant l'entraînement ne garantit pas que l'auto-encodeur soit incapable de reconstruire ces structures considérées comme anormales.

Une manière d'éviter ce problème est d'entraîner le réseau à projeter une image dont on a masqué une partie de celle-ci vers le domaine des images saines [13, 14]. Cette approche basée sur de l'*inpainting* comporte deux inconvénients principaux. Premièrement, afin que l'usage et l'entrainement du modèle soient les plus cohérents possible, il est préférable que le défaut soit contenu entièrement dans le masque à reconstruire. Deuxièmement, la reconstruction s'effectue par bloc et nécessite de déplacer le masque d'*inpainting* sur chacune des parties de l'image d'entrée. Afin de reconstruire l'image entière, de multiples inférences sont nécessaires, ce qui accroît le temps nécessaire pour effectuer la détection d'anomalie dans chaque image.

Dans notre approche, la corruption synthétique des images d'entraînement impose explicitement que la représentation latente soit invariante au bruit additif présent sur l'image d'entrée du réseau. [15]. Si l'ajout de bruit de type sel et poivre a déjà été utilisée [16], son bénéfice sur la reconstruction n'a pas été discuté. Dans notre section expérimentale, nous mettons en avant d'autres modèles de bruits qui sont, quant à eux, de forme spécifique et localisés à une partie de l'image. Nous montrons que de telles corruptions généralisent mieux aux défauts réels que les bruit appliqués selon une distribution aléatoire sur toute l'image. Cette robustesse apportée par une corruption synthétique soigneusement choisie rend possible l'ajout de skip-connections tout en évitant les désagréments liés à ceux-ci.

3 Méthode

Notre méthode s'appuie sur le résidu d'une reconstruction régularisée, c.-à-d. restreinte à l'espace des images saines, pour détecter les anomalies. Dans cette section, nous détaillons les différentes composantes de notre approche, allant de l'apprentissage de l'opérateur de régularisation à la détection d'anomalies par analyse du résidu de reconstruction.

3.1 Architecture du réseau convolutif

Afin de régulariser la reconstruction, nous avons choisi de travailler avec des réseaux convolutifs dont l'architecture se base sur celle d'un auto-encodeur. L'objectif de ces réseaux est de projeter l'image d'entrée $\mathbf{x} \in \mathcal{R}^{m \times n \times c}$ [1] vers un espace latent de plus faible dimension $\mathcal{R}^{m' \times n' \times c'}$ où $m' \times n' \times c' \ll m \times n \times c$. Cette projection, effectuée par la partie amont du réseau (encodeur) permet l'apprentissage d'une représentation compressée de l'image d'entrée. Ensuite, la partie aval du réseau (décodeur) effectue une projection depuis l'espace latent vers l'espace des images saines $\hat{\mathbf{x}} \in \mathcal{R}^{m \times n \times c}$. La régression s'opère sous la contrainte de minimisation de l'erreur de reconstruction au sens des moindres carrés

$$Loss(\mathbf{x}, \hat{\mathbf{x}}) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \sum_{k=0}^{c-1} \left(\mathbf{x}(i,j,k) - \hat{\mathbf{x}}(i,j,k)\right)^2 \quad (1)$$

Deux instances de réseaux ont été comparées :

Auto-Encodeur (AE). Le premier réseau est un auto-encodeur prenant en entrée une image de taille 256×256 et ayant un espace latent de dimension $4 \times 4 \times 512$. La projection vers l'espace de plus faible dimension se fait au moyen d'une succession de 6 couches convolutives avec un noyau de taille 5×5 et espacées par pas de 2 pixels [2]. Ensuite, 6 convolutions avec un noyau de taille 5×5 suivies d'un suréchantillonage de facteur 2 sont appliquées sur les vecteurs de l'espace latent afin de revenir dans l'espace original de l'image.

Auto-Encodeur avec *Skip-connections* (AESc). Le second réseau possède une architecture similaire au premier (AE). La différence réside dans l'ajout de *skip-connections* réalisant l'addition des vecteurs de l'encodeur à ceux du décodeur ayant la même dimension. Ce réseau est représenté sur la figure 2.

3.2 Paramètres d'entrainement

Les réseaux AE et AESc ont tous deux été entrainés pendant 250 *epochs* avec une taille de *batch* de 16 images. L'optimiseur employé est ADAM [17] avec un taux d'apprentissage de 0.005. Le réseau retenu pour l'inférence est sélectionné comme étant celui des 250 *epochs* minimisant la mesure de Peak Signal to Noise Ratio (PSNR) sur un validation set.
Pour chaque image d'entrainement, 5 nouvelles images sont générées à l'aide d'une corruption suivant un des quatre mécanismes décrits dans la section 3.3. À ces 5 images, sont ajoutées les 10 images correspondant à leur version modifiée via une symétrie horizontale et une verticale. Pour le jeu de données considéré (MVTec AD [18]), cela correspond à un nombre d'images compris entre 900 et 5865 selon la catégorie d'images.

3.3 Modèles de corruption

Pour rappel, l'auto-encodeur a pour tâche de préserver du mieux possible les structures saines tout en modifiant les structures anormales. De par l'impossibilité de collecter des paires d'images avec d'une part une image comprenant un défaut réel et d'autre part sa version saine, nous proposons de corrompre les images d'entrée à l'aide de défauts synthétiques. L'objectif de la corruption de l'image d'entrée est d'éviter la convergence du réseau vers l'opérateur identité et de favoriser l'apprentissage de représentations pertinentes pour les structures saines.

Nous proposons une comparaison de plusieurs types de défauts synthétiques :

1. Dans notre contexte applicatif qui se vise à portée industrielle, nous travaillons exclusivement avec des images en niveaux de gris. Cela implique que $c = 1$.

2. Se traduit en anglais par *stridées* d'un facteur 2.

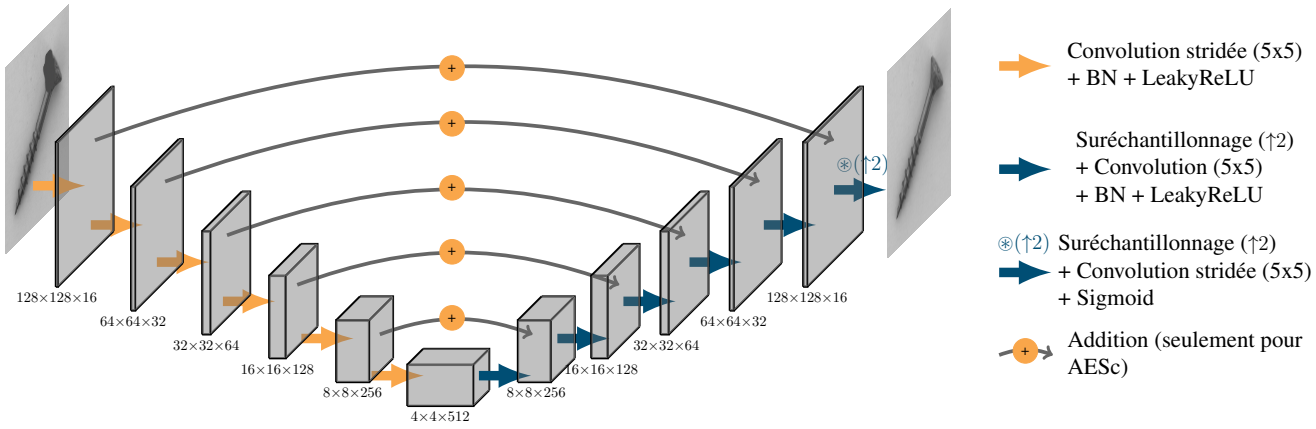

FIGURE 2 – Architecture de l'auto-encodeur AESc réalisant la reconstruction d'une image arbitraire de dimension 256×256 vers l'espace des images saines de dimension égale. Les dimensions des cartes de caractéristiques (*feature maps*) sont donnés par un triplet de valeurs : les deux premières correspondent à la dimension spatiale et la troisième à la profondeur des cartes de caractéristiques.

Bruit gaussien. Corruption de l'image par ajout de bruit blanc appliqué de manière uniforme sur toute l'image. Pour une image normalisée entre 0 et 1, la valeur corrompue d'un pixel x' correspondant à un pixel inital x est une réalisation d'une variable aléatoire suivant une distribution gaussienne de moyenne x et variance σ^2

$$x' = \mathcal{N}\left(x, \sigma^2\right) \tag{2}$$

où σ^2 peut prendre les valeurs $[0.1, 0.2, 0.4, 0.8]$.

Griffe. Corruption de l'image par ajout d'une courbe reliant deux points choisis de manière aléatoire dans l'image, et dont la couleur est choisie aléatoirement dans la dynamique des niveaux de gris. La courbe peut être une ligne droite, une forme sinusoïdale ou suivant la forme d'une fonction racine carrée.

Gouttes. Corruption de l'image par ajout d'une série de gouttelettes dont la couleur est choisie aléatoirement dans la dynamique des niveaux de gris. Chacune des 10 goulettes est circulaire et de diamètre variable (entre 1 et 2% de la plus faible dimension de l'image). Elles sont espacées de sorte à ce qu'elles se recouvrent partiellement.

Tache. Corruption de l'image par ajout d'une structure dont la couleur est choisie aléatoirement dans la dynamique des niveaux de gris et ayant une forme approximativement elliptique mais aux bords irréguliers. Plus concrètement, il s'agit de l'interpolation cubique de 20 points, ordonnés en ordre croissant de coordonnées polaires, aux alentours du contour d'une ellipse de taille (axes entre 1 et 12% de la plus faible dimension de l'image) et d'excentricité variables.

Plusieurs exemples de ces différentes corruptions sont visibles sur la figure 1.

3.4 Mécanisme de détection d'anomalies

Étant donné que le réseau a été entraîné pour prédire une image correspondant à la projection de l'image d'entrée sur la distribution des images saines, le caractère anormal de l'image d'entrée $\mathbf{x}$ peut être mesuré sur base du résidu de prédiction, défini comme la différence entre $\mathbf{x}$ et sa reconstruction régularisée $\hat{\mathbf{x}}$ (c.-à-d. carte résiduelle). Ainsi, un seuil peut être fixé sur la carte résiduelle afin d'identifier les pixels de l'image qui appartiennent ou non à la classe normale. Par la suite, nous ferons référence à cette méthode en tant que détection *pixel par pixel*.

Par ailleurs, la décision de normalité/anormalité peut être prise au niveau de l'image, appelée ici détection *image par image*, sur base de la norme de la carte résiduelle. Dans cet article, nous avons considéré l'étude d'une norme $\mathcal{L}^p$ définie par

$$\mathcal{L}^p(\mathbf{x}, \hat{\mathbf{x}}) = \left(\sum_{i=0}^{m} \sum_{j=0}^{n} |\mathbf{x}_{i,j} - \hat{\mathbf{x}}_{i,j}|^p \right)^{1/p} \tag{3}$$

avec $\mathbf{x}_{i,j}$ désignant le pixel de la $i^{\text{ème}}$ ligne et $j^{\text{ème}}$ colonne de l'image $\mathbf{x}$.
Quatre valeurs du paramètre p ont été considérées, à savoir 0, 1, 2 et ∞. Cependant, au vu des performances plus élevées obtenues pour $p = 2$ et $p = \infty$, seules les normes $\mathcal{L}^2$ et $\mathcal{L}^\infty$ seront présentées par la suite.

Comme mentionné précédemment, un seuil est nécessaire afin de décider de l'appartenance d'un pixel ou d'une image à la classe saine. Le calcul de celui-ci étant fortement dépendant des besoins liés à l'application, nous présenteront exclusivement les performances des différentes méthodes par l'aire sous la courbe (AUC) d'efficacité du récepteur (ROC), obtenue en balayant toutes les valeurs de seuils.

4 Résultats

Les expériences ont été réalisées sur les images de MVTec AD [18], constitué de 10 catégories d'objets et de 5 catégories de textures comportant des défauts réels. Ceux sont tous localisés grâce à un masque de segmentation obtenu manuellement pour les images de test. Toutes les images ont été mises à l'échelle de 256×256 pixels. La détection des défauts a été réalisée à cette résolution.
Le code est disponible à l'adresse `https://github.com/anncollin/AnomalyDetection-Keras`.

4.1 Analyse quantitative

Une analyse quantitative des résultats est fournie d'une part via la table 1 qui répertorie les performances obtenues pour la classification des images et d'autre part via la table 2 qui effectue une classification de chaque pixel.

Dans les deux cas, les méthodes présentées dans ce travail sont comparées à d'autres approches abordant également le problème par une approche basée sur la reconstruction régularisée. Ces méthodes ont été choisies pour le choix de leur démarche apportant une comparaison pertinente avec les nôtres.
Les méthodes basées sur la reconstruction des images arbitraires à l'aide d'un auto-codeur présentées par Bergmann et al. [7, 18] nous ont semblé pertinentes à titre comparatif. Cependant, les résultats que nous avons reproduits en interne sont inférieurs à ceux présentés dans les articles référencés. Nous avons donc décidé de nous comparer aux résultats originaux bien que ceux ci ne soient disponibles que pour la détection *pixel par pixel*. Une méthode distincte est ainsi utilisée pour la comparaison de notre méthode dans le cas d'usage de la détection *image par image*.

TABLE 1 – AUC de la détection *image par image* obtenus sur les différentes classes d'images en fonction de l'architecture de l'auto-encodeur et du modèle de bruit synthétique utilisé ("Identité" correspond aux images non altérées par un bruit synthétique). Pour nos méthodes (AE et AESc), la première ligne correspond à la détection via le calcul de la norme $\mathcal{L}^2$ et la deuxième via la norme $\mathcal{L}^\infty$. Les meilleures valeurs pour chaque catégorie sont mises en gras.

		AE	**AESc**							**ITAE [19]**
			Identité	Gouttes	Bruit gaussien				Griffe	Tache	
					$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.4$	$\sigma = 0.8$			
Textures	Carpet	0.43 0.54	0.48 0.35	0.80 0.88	0.44 0.75	0.50 0.61	0.45 0.60	0.59 0.63	0.77 0.86	0.79 **0.94**	0.71
	Grid	0.80 0.90	0.49 0.47	0.94 **0.98**	0.90 0.97	0.89 0.97	0.54 0.86	0.42 0.79	0.85 0.96	0.92 0.96	0.88
	Leather	0.41 0.87	0.42 0.71	0.84 0.93	0.59 0.89	0.64 0.89	0.66 0.90	0.64 0.81	0.79 0.93	0.91 **0.97**	0.87
	Tile	0.48 0.80	0.87 0.82	0.94 0.90	0.75 0.76	0.75 0.74	0.72 0.72	0.68 0.75	0.96 0.88	**0.98** 0.96	0.74
	Wood	0.95 0.84	0.86 0.83	**0.99** 0.98	0.84 0.82	0.89 0.84	0.87 0.88	0.88 0.88	0.94 0.85	0.96 0.97	0.92
Objets	Bottle	0.96 0.93	0.64 0.78	0.94 0.95	0.97 0.86	0.96 0.88	0.95 0.92	0.95 0.89	0.93 0.95	**0.98** 0.96	0.94
	Cable	0.65 0.50	0.42 0.41	0.53 0.63	0.65 0.43	0.6 0.40	0.58 0.37	0.53 0.42	0.55 0.60	**0.87** 0.84	0.83
	Capsule	0.71 **0.74**	0.65 0.58	0.66 0.70	0.65 0.71	0.53 0.70	0.39 0.69	0.42 0.70	0.67 0.71	**0.74** **0.74**	0.68
	Hazelnut	0.89 0.93	0.79 0.91	0.93 0.94	0.66 0.95	0.32 0.94	0.36 0.94	0.22 0.90	0.95 0.93	0.95 **0.96**	0.86
	Metal nut	0.55 0.52	0.24 0.34	0.24 0.43	0.53 0.59	0.57 0.54	0.76 0.57	0.79 0.55	0.26 0.47	0.84 **0.85**	0.67
	Pill	0.78 0.69	0.70 0.61	0.72 0.68	0.71 0.65	0.72 0.37	0.73 0.35	0.34 0.41	0.75 0.65	**0.81** 0.72	0.79
	Screw	0.83 0.76	0.38 0.69	0.91 0.76	**1.0** 0.83	0.98 0.84	0.95 0.96	0.01 0.97	0.54 0.79	0.58 0.76	**1.0**
	Toothbrush	0.95 0.95	0.66 0.70	0.89 0.94	0.95 0.83	0.76 0.68	0.87 0.67	0.78 0.66	0.83 0.84	0.98 **1.0**	**1.0**
	Transistor	0.77 0.74	0.50 0.46	0.78 0.79	0.80 0.63	0.73 0.67	0.57 0.66	0.45 0.58	0.71 0.76	0.87 **0.88**	0.84
	Zipper	0.81 0.76	0.58 0.70	0.78 0.82	**0.92** 0.77	0.80 0.76	0.75 0.73	0.72 0.76	0.85 0.84	0.90 0.83	0.80
	Moyenne	0.78 0.76	0.58 0.62	0.79 0.82	0.76 0.76	0.71 0.72	0.68 0.72	0.56 0.71	0.76 0.80	**0.88** **0.89**	0.84

TABLE 2 – AUC de la détection *pixel par pixel* obtenus sur les différentes classes d'images en fonction de l'architecture de l'auto-encodeur et du modèle de bruit synthétique utilisé. Les meilleures valeurs pour chaque catégorie sont mises en gras.

		AE	**AESc**								**AE (L2) [18]**	**AE (SSIM) [18]**
			Identité	Gouttes	Bruit gaussien				Griffe	Tache		
					$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.4$	$\sigma = 0.8$				
Textures	Carpet	0.54	0.53	0.64	0.58	0.57	0.56	0.52	0.64	0.68	0.59	**0.87**
	Grid	0.79	0.54	0.74	0.84	0.80	0.68	0.61	0.75	0.85	0.90	**0.94**
	Leather	0.77	0.69	0.87	0.69	0.52	0.48	0.51	0.83	**0.95**	0.75	0.78
	Tile	0.45	0.60	0.61	0.54	0.55	0.55	0.53	0.61	**0.70**	0.51	0.59
	Wood	0.63	0.62	0.70	0.65	0.65	0.65	0.63	0.70	**0.80**	0.73	0.73
Objets	Bottle	0.83	0.49	0.51	0.80	0.79	0.78	0.78	0.48	0.77	0.86	**0.93**
	Cable	0.58	0.67	0.58	0.51	0.45	0.38	0.36	0.66	0.82	**0.86**	0.82
	Capsule	0.81	0.57	0.61	0.38	0.33	0.28	0.29	0.64	0.80	0.88	**0.94**
	Hazelnut	0.92	0.81	0.79	0.57	0.55	0.64	0.58	0.82	0.89	0.95	**0.97**
	Metal nut	0.79	0.5	0.50	0.69	0.68	0.73	0.75	0.46	0.58	0.86	**0.89**
	Pill	0.80	0.64	0.66	0.46	0.43	0.50	0.75	0.66	0.70	0.85	**0.91**
	Screw	0.93	0.78	0.81	0.68	0.55	0.37	0.35	0.80	0.84	**0.96**	**0.96**
	Toothbrush	0.91	0.77	0.75	0.85	0.82	0.77	0.71	0.75	0.84	**0.93**	0.92
	Transistor	0.74	0.56	0.57	0.62	0.62	0.60	0.53	0.57	0.66	0.86	**0.90**
	Zipper	0.69	0.62	0.58	0.62	0.59	0.56	0.56	0.62	0.67	0.77	**0.88**
	Moyenne	0.75	0.63	0.66	0.63	0.59	0.57	0.56	0.67	0.77	0.82	**0.87**

Classification *image par image*. Une analyse détaillée de la table 1 révèle que la corruption des images lors de l'entraînement de AESc améliore significativement la détection des images comportant une anomalie. En particulier, un bruit structurel tel que Gouttes, Griffe ou Tache est plus efficace que Bruit gaussien.
Le modèle le plus performant est AESc + Tache qui surpasse presque systématiquement nos autres modèles ainsi que la méthode ITAE [19]. ITAE propose une reconstruction via un auto-encodeur muni de *skip-connections* et entraîné sur des images corrompues via des rotations et le passage en niveaux de gris des images en couleur. La détection est calculée sur basée de la norme $\mathcal{L}^1$ du résidu.
Il est aussi pertinent de mentionner que le modèle AE obtient de moins taux de détection que AESc + Tache.

Classification *pixel par pixel*. La prédominance de l'approche AESc + Tache n'est par contre pas confirmée par la table 2 puisque les méthodes de l'état de l'art offrent de meilleures performances. Les modèles AE (L2) et AE (SSIM) tels que décrits dans [7] et [18] sont tous deux des auto-encodeurs sans *skip-connections* entraînés pour minimiser l'erreur de reconstruction sur des images non-corrompues, soit au sens des moindres carrés (L2), soit au sens de la Structural SIMilarity (SSIM). Ces méthodes sont donc similaires dans leur principe à notre méthode AE, en particulier lorsque l'erreur de reconstruction est mesurée par la norme L2. On peut donc s'étonner de la différence de performance observée entre la colonne AE et la colonne AE (L2). Elle s'explique en partie par le fait que AE (L2) reconstruit les textures par patch de 128×128, tandis que notre méthode AE travaille sur des images complètes à la résolution 256×256, à la fois pour les objets et les textures. Par ailleurs, l'augmentation des données ainsi que l'architecture de l'auto-encodeur sont également différentes. Cependant, ni [7] ni [18] ne mettent leur code à disposition, ce qui rend la comparaison exacte difficile. Nous pensons cependant que la comparaison entre AE et AESc est la plus pertinente, dans la mesure où ces deux approches adoptent des réseaux qui ne diffèrent que par l'absence ou la présence de *skip-connections*, et sont entraînés de manière comparable.

Nous mettons l'emphase sur deux observations intermédiaires qui seront commentées plus en détail par la suite. Premièrement, malgré le fait que l'état de l'art offre de meilleures performances générales, le modèle AESc + Tache obtient les meilleurs taux de classification des pixels défectueux pour 3 des 5 catégories de texture. Deuxièmement, bien que légèrement inférieures, les performances moyennes de AESc + Tache sont relativement proches de celles obtenues par le modèle AE. Plus précisément, AESc + Tache est plus performant que AE sur les textures alors que la tendance inverse est observée pour les objets.

4.2 Analyse qualitative du modèle de bruit synthétique

L'analyse visuelle d'un exemple représentatif d'un objet ainsi que d'une texture (figure 3) permet de soulever des comportements communs à l'ensemble du jeu de données. Sans surprise, AESc (Identité) illustre la tendance du modèle à converger vers l'opérateur identité. La corruption des images d'entraînement atténue cet effet de façon significative. Par ailleurs, la nature du bruit synthétique a un effet sur la reconstruction. On peut ainsi observer que les bruits structurels (Gouttes, Griffe et Tache) ont tendance à préserver les structures saines tout en ne modifiant que les zones de l'image correspondant à une anomalie. A l'inverse, lorsque le bruit est appliqué de

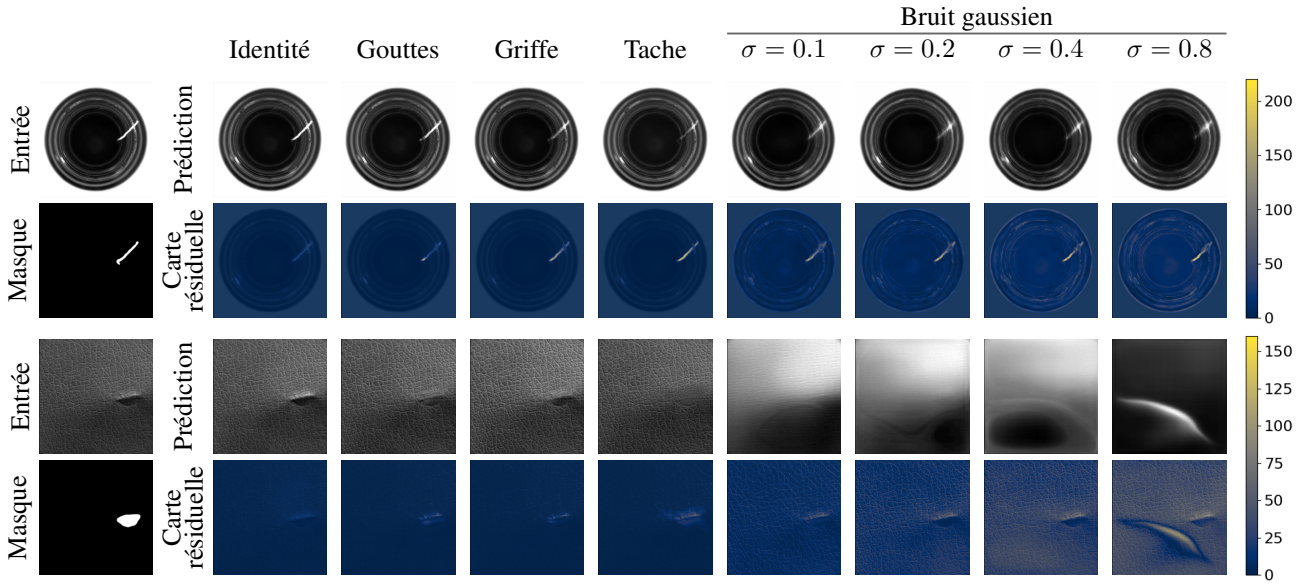

FIGURE 3 – Impact du modèle de corruption sur la reconstruction d'un object défectueux (haut) (classe Bottle) et d'une texture défectueuse (bas) (classe Leather) avec AESc.

manière uniforme sur toute l'image via le modèle de Bruit gaussien, l'ensemble des structures de l'image ont tendance à être modifiées, et ce, indépendemment de leur nature normale ou anormale. Nous noterons également la difficulté de la tâche de reconstruction de textures fines lorsqu'un bruit gaussien est appliqué sur les images d'entraînement. Seul un bruit structurel combiné à l'ajout de *skip-connections* semble produire une reconstruction précise des structures fines.

Nous faisons remarquer que pour les deux exemples, la reconstruction qui est la plus semblable aux images saines est produite par modèle AESc + Tache. Ce résultat est d'autant plus remarquable sur l'objet car même si le défaut a l'apparence d'une griffe, le modèle AESc + Tache génère une reconstruction plus adéquate que AESc + Griffe.

4.3 Analyse qualitative de l'ajout de bruit synthétique et de *skip-connections*

La tendance du réseau AESc à converger vers l'opérateur identité réside dans la présence des *skip-connections*. Pour la détection d'anomalies, ce phénomène peut être indésirable puisque la non modification des zones anormales de l'image ne peut mener à leur détection. Il est ainsi pertinent de vouloir comparer le comportement du réseau AESc à celui de AE. Pour la suite de l'analyse, nous avons décidé de montrer exclusivement les résultats obtenus avec AESc sans corruption ainsi qu'avec le bruit synthétique menant aux meilleurs résultats, à savoir le modèle Tache.

Supériorité de AESc + Tache pour les textures. Les reconstructions pour un exemple type de texture saine et défectueuses (figure 4) obtenues avec le modèle AE sont relativement floues. L'erreur de reconstruction est globalement plus élevée pour la texture défectueuse mais la localisation du défaut n'est pas réalisable hors de cette carte résiduelle. Comme attendu, le modèle AESc sans corruption tend vers un opérateur identité rendant toute détection d'anomalies impossible. Le modèle AESc, quant à lui, parvient à assurer une reconstruction fine des structures saines tout en modifiant les régions défectueuses de sorte à ce qu'elles puissent être identifiés comme telles. Bien que la reconstruction des défauts de AESc ne soit pas visuellement satisfaisante pour tromper l'oeil humain, la différence est suffisante pour la détection des régions anormales.

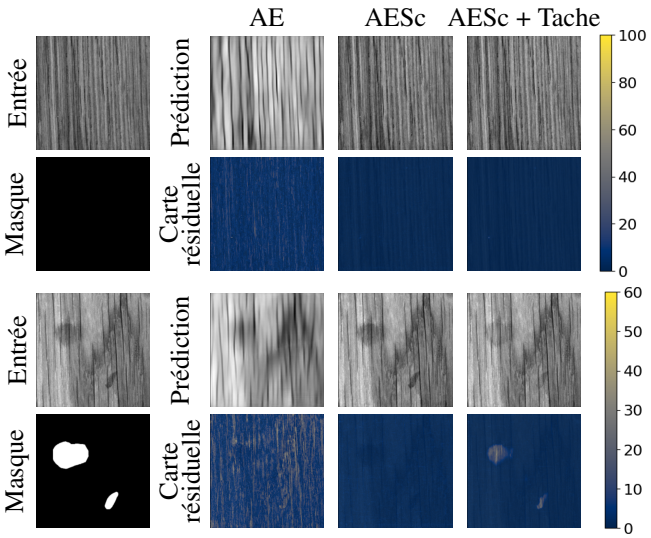

FIGURE 4 – Reconstruction d'une image saine (haut) et défectueuse (bas) issues de la catégorie de textures Wood.

Résultat positif de AE sur un objet. Si la tendance générale du modèle AE à modifier l'apparence globale de l'image est indésirable pour les textures, elle peut cependant être avantageuse pour la détection d'anomalies dans des objets. Pour illustrer ce propos, nous avons choisi deux exemples pour lesquels le modèle AESc + Tache obtient de meilleurs taux de détection *image par image* que le modèle AE, mais de moins bons taux pour la détection *pixel par pixel*.

L'exemple du haut de la figure 5, montre une pièce défectueuse dont le défaut est situé sur l'ensemble de sa surface. Le modèle AE se montre plus invariant par rapport aux distorsions de l'image d'entrée. Pour cette catégorie, ce modèle produit une image floue presque identique pour toutes les images du jeu de test. Par ailleurs, le modèle AESc produit une reconstruction plus fidèle à l'image d'entrée mais qui rend la détection des pixels défectueux incomplète.

L'exemple du bas de la même figure montre un transistor pour lequel la puce devrait se situer au centre de l'image et les pins au bas de cette dernière. Les modèles AE et AESc + Tache reconstruisent tous deux une image reproduisant les structures les plus invariantes vues durant l'entraînement (c.-à-d. les bords de l'image). Cependant, la carte résiduelle de AE semble mieux recouvrir la zone correspondant au défaut que celle de AESc.

Ces deux exemples sont une illustration de la raison pour laquelle on observe une baisse des performances entre la détection *image par image* et *pixel par pixel* pour le modèle AESc + Tache.

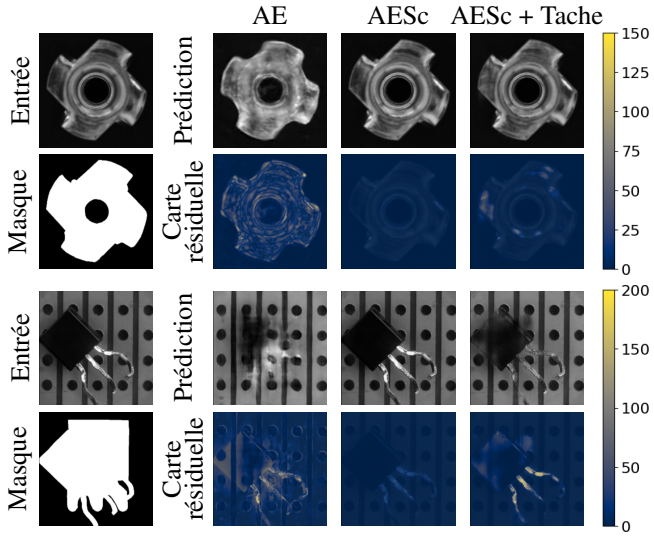

FIGURE 5 – Reconstruction d'un object défectueux de la classe Metal nut (haut) et Transistor (bas) pour lequel le défaut se situe sur l'ensemble de la pièce.

Résultat positif de AESc + Tache sur un objet. Si les deux exemples illustrés précédemment tendent à suggérer que l'invariance vis à vis de l'image d'entrée du modèle AE est désirable pour la détection d'anomalies sur des objets, le propos reste toutefois à nuancer.
Lorsque le défaut est présent sur une petite partie de l'image, la conservation des structures normales, telle que favorisée par les *skip-connections* présentes dans le réseau AESc, reste d'une importance cruciale. Ainsi, comme illustré sur la figure 6, à la fois le modèle AE et AESc montrent une erreur de reconstruction plus élevée à l'endroit du défaut. Cependant, AE altère également deux autres régions saines de l'image. Bien que le défaut soit visuellement mieux reconstruit par le modèle AE, il est plus facilement identifiable à partir la carte résiduelle obtenue par le modèle AESc + Tache.

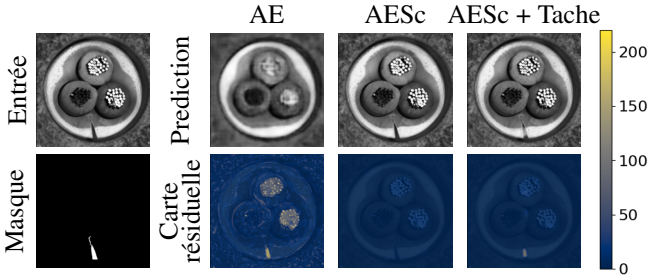

FIGURE 6 – Reconstruction d'un objet défectueux de la classe Cable pour lequel le défaut est localisé sur une partie de la pièce.

5 Conclusion

Dans cet article, nous avons montré l'efficacité d'une approche visant à reconstruire une image saine à partir d'une image arbitraire afin de s'appuyer sur le résidu de reconstruction pour déceler les éventuelles anomalies de l'image de départ. A cette fin, des auto-encodeurs entraînés exclusivement à partir d'images dérivées de données saines ont été utilisés. Plus précisément, nous avons montré les avantages apportés par l'ajout simultané de *skip-connections* et de corruption lors de l'entraînement du réseau.
Si d'une part ces deux éléments parviennent à améliorer considérablement la détection des images défectueuses, leur intérêt pour la segmentation des régions anormales est plus nuancé. D'une part, la reconstruction obtenue a tendance à être plus fidèle à l'image d'origine lorsque ces deux modifications sont appliquées. Dans les cas où le défaut est l'objet dans son entièreté, ces modifications mènent ainsi à de moins bonnes prédictions de la région anormale. Par contre, lorsque le défaut est une zone plus restreinte d'une texture ou d'un objet, la capacité du modèle à reconstruire les structures saines de manière très fidèle engendre une meilleure localisation de la région anormale de l'image.

Références

[1] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections. *CoRR*, abs/1606.0 :1–17, 2016.

[2] Marco A.F. Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal Processing*, 99 :215–249, 2014.

[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Survey of Anomaly Detection. *ACM Computing Survey*, (September) :1–72, 2009.

[4] Paolo Napoletano, Flavio Piccoli, and Raimondo Schettini. Anomaly detection in nanofibrous materials by CNN-based self-similarity. *Sensors (Switzerland)*, 18(1), 2018.

[5] Benjamin Staar, Michael Lütjen, and Michael Freitag. Anomaly detection with convolutional neural networks for industrial surface inspection. *Procedia CIRP*, 79(January 2019) :484–489, 2019.

[6] Chong Zhou and Randy C. Paffenroth. Anomaly Detection with Robust Deep Autoencoders. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 665–674, 2017.

[7] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *VISIGRAPP 2019 - Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 5 :372–380, 2019.

[8] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10265 LNCS :146–147, mar 2017.

[9] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially Learned One-Class Classifier for Novelty Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3379–3388, 2018.

[10] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-AnoGAN : Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 54(January) :30–44, 2019.

[11] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain MR images. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11383 LNCS :161–169, 2019.

[12] Samet Akçay, Amir Atapour-Abarghouei, and Toby P. Breckon. Skip-GANomaly : Skip Connected and Adversarially Trained Encoder-Decoder Anomaly Detection. *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1—-8, 2019.

[13] Matthias Haselmann, Dieter P. Gruber, and Paul Tabatabai. Anomaly Detection Using Deep Learning Based Image Completion. *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*, pages 1237–1242, 2019.

[14] Asim Munawar and Clement Creusot. Structural inpainting of road patches for anomaly detection. *Proceedings of the 14th IAPR International Conference on Machine Vision Applications, MVA 2015*, pages 41–44, 2015.

[15] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders : Explicit invariance during feature extraction. *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, (1) :833–840, 2011.

[16] Shuang Mei, Yudan Wang, and Guojun Wen. Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model. *Sensors (Switzerland)*, 18(4) :1–18, 2018.

[17] Diederik P. Kingma and Jimmy Lei Ba. Adam : A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15, 2015.

[18] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. *Cvpr 2019*, pages 9592–9600, 2019.

[19] Chaoqing Huang, Jinkun Cao, Fei Ye, Maosen Li, Ya Zhang, and Cewu Lu. Inverse-Transform AutoEncoder for Anomaly Detection. *arXiv preprint arXiv :1911.10676*, 2019.