Normalizing Flows contrastifs pour la détection d'anomalies sur ouvrages d'art

Brice MARC[1,2] Philippe FOUCHER[1] Florence FORBES[2] Pierre CHARBONNIER[1]

[1]Équipe de recherche ENDSUM, Cerema, Strasbourg

[2]Équipe-projet STATIFY, Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, Grenoble

Résumé – Les méthodes via *Normalizing Flows* (NF) sont des méthodes de détection d'anomalies non supervisées qui performent bien dans le cadre de l'auscultation des ouvrages d'art (OA). Utilisant uniquement des images sans défauts, elles apprennent à détecter les anomalies comme les éléments se démarquant de la distribution des parties saines. Dans ces travaux, nous mettons en place différents objectifs d'apprentissage complémentaires utilisant des anomalies synthétiques, générées par interpolation de POISSON, afin d'améliorer les performances du modèle NF CFlow-AD, un des meilleurs modèles NF de l'état de l'art. Nous montrons l'intérêt de ces objectifs sur plusieurs jeux de données relatifs aux OA.

Abstract – Among unsupervised anomaly detection methods in the context of civil engineering (CE) monitoring, those using Normalizing Flows (NF) have reached state-of-the-art performance. Using only defect-free images, they learn to detect anomalies as elements departing from the healthy parts distribution. In this work, we propose to increase the discriminative power of these methods by leveraging the possibility to produce synthetic anomalies. Starting with CFlow-AD, one of the best-performing NF-based methods, we augment its loss with different complementary learning objectives using anomalies generated by POISSON interpolation. We demonstrate the interest of these new augmented losses on several CE-related datasets.

1 Introduction

L'auscultation des ouvrages d'art est une tâche indispensable pour assurer leur durabilité et leur sécurité. Elle nécessite de relever les anomalies présentes sur les infrastructures, procédé fastidieux qui demande du temps. Ainsi, des méthodes fondées sur le traitement de l'image et l'apprentissage profond se développent pour automatiser cette tâche. Les méthodes les plus récentes s'appuient sur l'apprentissage non supervisé. Les modèles sont, dans ce cas, entraînés uniquement avec des images sans défauts, dites « saines ». L'apprentissage non supervisé permet ainsi de s'affranchir du coût de la production d'annotations, nécessaires avec les méthodes supervisées traditionnelles.

Parmi les différents paradigmes d'apprentissage non supervisé [1], les méthodes utilisant des *Normalizing Flows* (NF, Sec. 2.1), se montrent performantes sur les images d'ouvrages d'art [2]. Un NF est un modèle chargé d'apprendre une transformation bijective entre deux distributions. Dans le cadre de la détection d'anomalies non supervisée, le modèle apprend à transformer la distribution des images saines du jeu d'entraînement vers une distribution gaussienne multivariée. Ensuite, lorsque la transformation apprise est appliquée sur de nouvelles images pouvant contenir des anomalies, le modèle détecte comme défauts les parties des images qui sont projetées en dehors de la distribution.

Nous proposons ici d'améliorer la capacité de la fonction de perte utilisée dans un modèle NF à séparer encore mieux la distribution des parties saines de celle des anomalies. Pour cela, nous générons des anomalies synthétiques (Sec. 2.2) et nous ajoutons à la fonction de perte des objectifs d'apprentissage supplémentaires, fondés sur un principe d'apprentissage contrastif (*contrastive learning*, Sec. 2.3). Nous testons l'influence de différents objectifs contrastifs sur un modèle NF de l'état de l'art, entraîné sur des jeux de données relatifs aux ouvrages d'art (Sec. 3).

2 Méthodes

2.1 Modèle fondé sur des Normalizing Flows

CFLOW-AD [3], une méthode non supervisée via *Normalizing Flows* (NF) parmi les plus performantes de l'état de l'art, est appliquée pour réaliser la détection d'anomalies.

Dans un premier temps, un réseau convolutionnel, appelé extracteur, pré-entraîné sur le large jeu de données ImageNet [4], est utilisé pour extraire des caractéristiques (*features*) des images, à trois niveaux de résolution. Les pondérations de l'extracteur ne sont pas mis à jour lors de l'apprentissage.

Une fois les trois étages de *features* extraits, ils sont parallèlement traités par trois NF de même structure, mais ayant des pondérations propres. Chaque NF est composé d'une suite de huit *Affine Coupling Layers* [5]. Ces structures, de paramètres θ, permettent d'apprendre une transformation bijective de la distribution des cartes de *features* $x \in \mathcal{X}$ vers une distribution choisie de forme simple, tout en calculant le jacobien de cette transformation. Ainsi, en notant ϕ_θ la transformation, p_θ la distribution des cartes de *features* saines et $p_\mathcal{Z}$ la distribution simple, la log-vraisemblance de x peut s'écrire :

$$\log p_\theta(x) = \log p_\mathcal{Z}(\phi_\theta(x)) + \log |\det J_{\phi_\theta}(x)|. \quad (1)$$

Ainsi, en entraînant le NF pour apprendre une transformation vers une distribution $p_\mathcal{Z}$ gaussienne multivariée centrée-réduite, en dimension d, l'équation (1) devient :

$$\log p_\theta(x) = \frac{d}{2}\log(2\pi) - \frac{1}{2}\left\|\phi_\theta(x)\right\|^2 + \log |\det J_{\phi_\theta}(x)|. \quad (2)$$

Dans le cadre d'un apprentissage purement non supervisé, le modèle apprend uniquement la transformation des *features* issues d'images saines vers la distribution normale. Il faut donc maximiser la log-vraisemblance des projections, ce qui est équivalent à minimiser son opposée. L'objectif d'apprentissage est donc :

$$\mathcal{L}_{\text{NF}} = \mathbb{E}_{x\in\mathcal{X}}\left[\frac{1}{2}\left\|\phi_\theta(x)\right\|^2 - \log\left|\det J_{\phi_\theta}(x)\right|\right]. \quad (3)$$

Pour, ensuite, déterminer la présence d'anomalies lors d'une inférence, on peut calculer un score exprimé en fonction de la log-vraisemblance, défini par l'expression (4). Le score étant calculé pour chacune des *features* des trois étages, un score pour chacun des pixels de l'image peut être défini comme le produit des scores des trois étages, remis à l'échelle de l'image. Un score global par image peut être défini comme le score maximal parmi les scores de chaque pixel.

$$S(x) = 1 - \exp(\log p_\theta(x)). \quad (4)$$

Bien qu'un apprentissage n'utilisant que des données saines puisse se montrer performant, nous proposons d'utiliser des anomalies synthétiques via un apprentissage contrastif afin d'augmenter le pouvoir de discrimination du modèle.

2.2 Anomalies synthétiques

Les anomalies synthétiques utilisées dans la littérature sont généralement obtenues par collage sur l'image cible, d'une autre image disparate, en suivant un masque aléatoire, comme proposé dans [6]. De telles anomalies peuvent avoir un aspect assez éloigné de la réalité.

Pour générer des anomalies réalistes, nous proposons un procédé de synthèse d'anomalies fondé sur l'interpolation de POISSON [7]. Celle-ci permet de coller une partie d'une image source sur une image cible, en évitant toute discontinuité au bord du collage. En notant f^* l'image cible et Ω le domaine d'interpolation, résoudre l'interpolation de POISSON revient à trouver numériquement l'interpolant f respectant les équations suivantes :

$$f = \underset{f}{\operatorname{argmin}} \iint_\Omega |\nabla f - v|^2 \text{ avec } f|_{\partial\Omega} = f^*|_{\partial\Omega}, \quad (5)$$

$$\Delta f = \operatorname{div} v \text{ sur } \Omega. \quad (6)$$

où v est un champ vectoriel pouvant être choisi comme le gradient de l'image source, ou comme un champ dépendant aussi du gradient de l'image cible. La première option sera retenue car elle est la meilleure pour coller des éléments opaques. Certains travaux comme [8] ont déjà employé l'interpolation de POISSON dans le cadre de la détection d'anomalies, mais ils utilisent d'autres images saines comme images sources, et produisent donc des défauts peu vraisemblables. Afin de générer des images réalistes, nous utilisons des anomalies provenant d'autres jeux de données en interpolant la partie correspondant à l'anomalie.

2.3 Méthodes par apprentissage contrastif

Pour chaque image du jeu d'entraînement, une anomalie synthétique est interpolée. Les images réelles et augmentées sont alors mises en opposition suivant plusieurs méthodes, afin d'apprendre au modèle à mieux séparer les distributions des parties saines et des défauts. On notera $\mathcal{X}'$ l'ensemble des cartes de *features* des images augmentées, $(x, x') \in \mathcal{X} \times \mathcal{X}'$ un couple de cartes de *features* correspondantes entre une image réelle et son image augmentée associée et $M_{x'}$ le masque indiquant quelles *features* de la carte décrivent l'anomalie.

Le modèle n'utilisant pas d'image augmentée est choisi comme méthode de référence. Sa fonction de perte est :

$$\mathcal{L}_{\text{None}} = \mathcal{L}_{\text{NF}}.$$

BGAD [9] est une méthode faiblement supervisée qui utilise quelques échantillons contenant des défauts, provenant du jeu de test, pour augmenter le pouvoir discriminant d'un modèle NF. Cette méthode vise à écarter la distribution des transformations des *features* relatives aux parties saines des images, de celles relatives aux anomalies. Cette méthode peut être applicable en remplaçant les vrais échantillons par des anomalies synthétiques. Pour écarter les distributions, le quantile $q(\beta)$ (β : hyperparamètre fixé à 5%) de la distribution des log-vraisemblances des *features* saines est choisi comme frontière. Les log-vraisemblances saines inférieures à $q(\beta)$ ainsi que les log-vraisemblances des anomalies supérieures à $q(\beta) - \tau$ (τ : hyperparamètre fixé à 0.1) sont pénalisées. Ainsi, l'objectif d'apprentissage est :

$$\mathcal{L}_{\text{boundary}} = \mathcal{L}_{\text{NF}} + \mathbb{E}_{x'\in\mathcal{X}'}\Big[\big\|(1 - M_{x'}).\min\big(\log p_\theta(x') - q(\beta),\ 0\big)\big\| + \big\|M_{x'}.\max\big(\log p_\theta(x') - q(\beta) + \tau,\ 0\big)\big\|\Big].$$

La méthode BGAD ne bénéficie cependant pas du fait que les images augmentées correspondent aux images saines avec une anomalie interpolée. Des approches fondées sur les différences entre une image et sa version augmentée peuvent être envisagées. Une méthode inspirée du modèle CL-Flow [10], consiste à comparer les transformations de la paire d'images faites par le modèle NF. Le but est de rapprocher les transformations des *features* relatives aux parties saines des deux images, tout en éloignant les *features* relatives à l'anomalie synthétique de celles aux mêmes emplacements dans l'image saine, à l'aide de la *cosine similarity*, CosSim. L'objectif d'apprentissage est le suivant :

$$\mathcal{L}_{\text{CL}} = \mathcal{L}_{\text{NF}} + \mathbb{E}_{(x,x')\in\mathcal{X}\times\mathcal{X}'}\Big[\mathrm{CosSim}(M_{x'}.\phi_\theta(x), M_{x'}.\phi_\theta(x')) - \mathrm{CosSim}((1 - M_{x'}).\phi_\theta(x), (1 - M_{x'}).\phi_\theta(x'))\Big].$$

Toutefois, cette méthode ne considère que les transformations des *features* et n'utilise pas le jacobien, pourtant important dans le calcul de la vraisemblance et pour détecter les anomalies. Ainsi, toujours dans l'optique de rapprocher les *features* saines entre elles et d'éloigner les *features* des anomalies de synthèse de celles des images réelles, un nouvel objectif fondé sur la log-vraisemblance est défini par :

$$\mathcal{L}_{\text{Likelihood}} = \mathcal{L}_{\text{NF}} + \mathbb{E}_{(x,x')\in\mathcal{X}\times\mathcal{X}'}\Big[\big\|(1 - M_{x'}).(\log p_\theta(x) - \log p_\theta(x'))\big\| - \big\|M_{x'}\big(\log p_\theta(x) - \log p_\theta(x')\big)\big\|\Big].$$

Ces deux dernière méthodes exploitent bien les différences entre une image réelle et sa version augmentée, mais perdent l'aspect global que la méthode BGAD propose. Ainsi, une dernière méthode combinant les objectifs est testée :

$$\mathcal{L}_{\text{fusion}} = \mathcal{L}_{\text{boundary}} + \mathcal{L}_{\text{Likelihood}}$$

3 Expérimentations

3.1 Jeux de Données

Afin de pouvoir évaluer les performances de chaque approche, des expériences sont menées sur différents jeux de données.

Tout d'abord, les méthodes sont appliquées sur une catégorie du *benchmark* MVTec [11]. MVTec est un jeu de données construit pour évaluer les algorithmes de détection non supervisés sur des images industrielles. Ces images sont réparties en 15 catégories et représentent des objets ou des surfaces. La catégorie surfacique *Tuile* est celle qui se rapproche le plus des images d'ouvrage d'art (OA), notre cas d'application, de par les motifs irréguliers qu'elle présente. Les différentes méthodes seront donc testées sur cette catégorie. Le jeu d'entraînement est constitué de 230 images saines et le jeu de test contient 33 images saines et 84 images avec des anomalies. Afin de générer des anomalies synthétiques, les images avec défauts de la catégorie *Bois* de MVTec sont utilisées comme images sources.

Les algorithmes de détection se comportent particulièrement bien sur les images de MVTec, qui sont prises sous des milieux où l'éclairage, la distance et l'angle d'acquisition sont très contrôlés. Ces conditions diffèrent de celles des images d'OA, beaucoup moins homogènes. Ainsi, pour évaluer la transférabilité des méthodes à notre cas d'application, un jeu d'images de chaussées et deux jeux spécifiques aux OA sont employés.

Crack500 [12] est un jeu de données regroupant 500 images de chaussées fissurées, réparties équitablement en un jeu d'entraînement et un jeu de test. Les 150 premières images du jeu d'entraînement sont découpées en 9 sous-images et triées pour obtenir 613 images saines, qui constitueront notre jeu d'entraînement. Les 100 dernières images du jeu d'entraînement sont aussi découpées en 9 pour obtenir le jeu de test, constitué de 321 images saines et de 527 images comportant des fissures. Les anomalies synthétiques sont générées à l'aide d'un deuxième jeu de données, DeepCrack [13], composé lui aussi d'images de chaussées fissurées.

Deux autres jeux de données ont été constitués grâce à des acquisitions que nous avons menées sur des ouvrages. Le premier ouvrage est un tunnel routier situé près de la commune Rive-de-Gier (*RDG*), comportant des fers apparents. 1003 images saines constituent le jeu d'entraînement. Le jeu de test comporte 481 images saines et 322 images avec défauts. Le second ouvrage est un passage inférieur routier situé dans la commune de Lingolsheim (*Lingo*). Les parois comportent des fers apparents (morceaux de béton qui tombent et laissent les tiges métalliques apparentes), des fissures et des pores. 107 images saines composent le jeu d'entraînement. Le jeu de test est divisé en 114 images avec défauts et 4 images saines. Les images qui sont utilisées pour générer des anomalies synthétiques sur ces deux jeux sont des fers apparents d'images d'OA du jeu de données CODEBRIM [14].

3.2 Paramètres choisis

Le réseau convolutionnel choisi comme extracteur de *features* est le réseau EfficientNet B6 [15]. Ce réseau s'est montré plus performant sur les jeux de données relatifs aux OA que le réseau Wideresnet50, réseau habituellement utilisé dans les travaux de détection d'anomalies.

Les images d'entrée sont redimensionnées en images de 256×256 pixels et sont regroupées par batchs de 8 images.

Le *learning rate* initial est de $2e^{-4}$ et est divisé par 10 après les époques 50, 75 et 90, sur une durée totale d'apprentissage de 200 époques. Les 8 premières époques ne font pas intervenir les fonctions de perte (*losses*) contrastives.

3.3 Métriques d'évaluation

Nous employons des métriques de type AUROC, relatives à l'aire sous la courbe ROC (taux de vrais positifs en fonction du taux de faux positifs), car elles sont indépendantes du choix d'un seuil de détection. La métrique *Pix.AUROC* est l'aire sous la courbe ROC, pour une classification de chaque pixel. La métrique *Pix.AUPRO* est une métrique locale similaire, mais pondère la taille des anomalies, pour que les grosses anomalies aient tout autant d'impact que les plus fines. Les scores d'un modèle parfait seraient de 100% tandis que ceux d'un modèle aléatoire seraient de 50%.

3.4 Résultats

Les résultats obtenus sur chaque jeu de données et pour chacun des objectifs d'apprentissage sont résumés dans le tableau 1. Des exemples de détections sont présentés dans la figure 1.

Loss	*Tuile*	*Crack500*	*RDG*	*Lingo*
$\mathcal{L}_{\text{None}}$	96.48 / 91.39	94.65 / 73.68	91.56 / 76.96	**93.56** / **78.24**
$\mathcal{L}_{\text{boundary}}$	<u>97.74</u> / <u>94.47</u>	**95.30** / <u>76.54</u>	**92.03** / **78.58**	93.33 / 77.36
$\mathcal{L}_{\text{CL}}$	96.3 / 90.47	94.42 / 71.60	90.98 / 75.03	93.53 / 78.00
$\mathcal{L}_{\text{likelihood}}$	97.48 / 93.89	95.16 / 75.79	91.41 / 77.59	<u>93.55</u> / <u>78.22</u>
$\mathcal{L}_{\text{fusion}}$	**97.76** / **94.53**	<u>95.23</u> / **76.62**	<u>91.81</u> / <u>78.31</u>	93.25 / 77.21

TABLEAU 1 : Performances de segmentation en fonction de la *loss* contrastive utilisée (scores *Pix AUROC* / *Pix AUPRO*, exprimés en $\%$) pour chaque jeu de données, **meilleurs résultats** et <u>2^{eme} meilleurs résultats</u> indiqués

Les apprentissages réalisés avec les objectifs $\mathcal{L}_{\text{boundary}}$ et $\mathcal{L}_{\text{fusion}}$ se placent en première ou deuxième position sur tous les jeux de données, à l'exception de *Lingo*. Puisque ces objectifs comprennent la *loss* $\mathcal{L}_{\text{boundary}}$, on peut en déduire que c'est celle-ci qui permet le mieux au modèle d'intégrer les anomalies synthétiques sur les jeux de données *Tuile*, *Crack500* et *RDG*. En effet, elle permet d'affiner les détections en proposant des contours plus précis qu'avec $\mathcal{L}_{\text{None}}$ ou $\mathcal{L}_{\text{CL}}$.

Les résultats sur le jeu de données *Lingo* se démarquent car le modèle sans apprentissage contrastif, $\mathcal{L}_{\text{None}}$ se montre le plus performant. Les objectifs contrastifs locaux $\mathcal{L}_{\text{CL}}$ et $\mathcal{L}_{\text{Likelihood}}$ permettent tout de même d'obtenir des résultats très proches et l'approche globale $\mathcal{L}_{\text{boundary}}$ dégrade légèrement les performances. Il est toutefois visuellement difficile d'observer des différences entre les prédictions obtenues suivant les objectifs considérés. Les anomalies synthétiques générées sur le jeu de données sont peut être trop éloignées des anomalies réelles, qui ont des tailles et des aspects très variables sur ce jeu. Cela

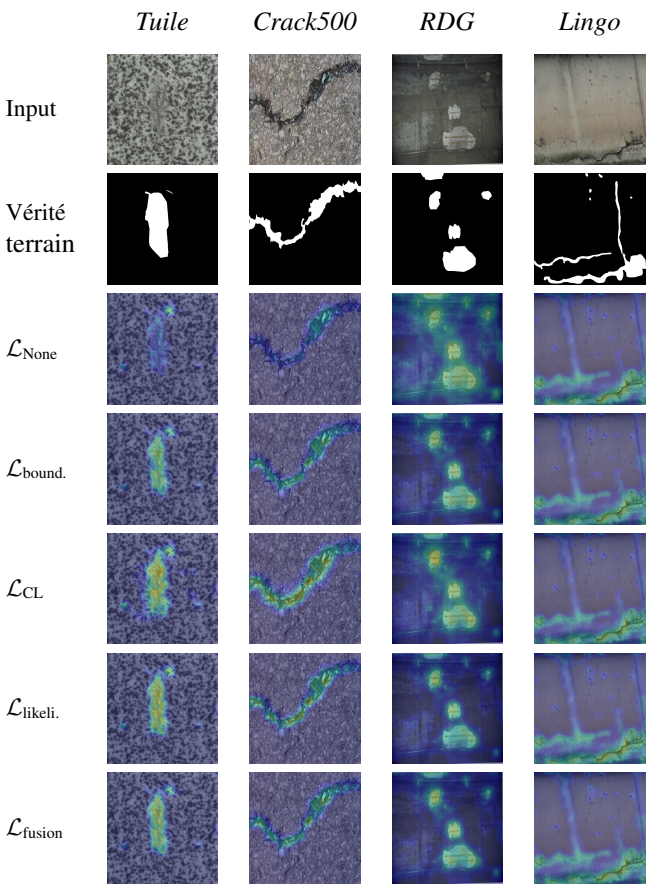

FIGURE 1 : Exemples de résultats obtenus avec chacune des fonctions de perte contrastives

ne permettrait pas aux apprentissages contrastifs d'améliorer les performances du modèle.

4 Conclusion

L'ajout d'un mécanisme d'apprentissage contrastif avec anomalies synthétiques dans un algorithme de *Normalizing Flow* a un effet positif sur la qualité de détection de défauts.

L'utilisation de la fonction objectif $\mathcal{L}_{\text{boundary}}$ a permis d'améliorer les performances du modèle sur les jeux de données *Tuile*, *Crack500* et *RDG*. Cet objectif global est plus adapté que les objectifs locaux $\mathcal{L}_{\text{CL}}$ et $\mathcal{L}_{\text{likelihood}}$, moins précis sur les contours des détections.

Sur le jeu de données *Lingo*, l'absence d'impact de l'apprentissage contrastif, qui dégrade même très légèrement les performances, vient toutefois questionner l'influence de l'apparence des anomalies de synthèse et de leur ressemblance avec les anomalies réelles. Une étude devra être menée pour caractériser l'effet de la taille, de la forme ou encore de la texture des anomalies sur l'efficacité des objectifs contrastifs.

5 Remerciements

Ces travaux ont bénéficié d'un accès aux moyens de calcul de l'IDRIS au travers de l'allocation de ressources AD011014508R1 attribuée par GENCI.

Références

[1] Y. Cui, Z. Liu, and S. Lian. A Survey on Unsupervised Anomaly Detection Algorithms for Industrial Images. *IEEE Access*, 2023.

[2] B. Marc, P. Foucher, F. Forbes, and P. Charbonnier. Évaluation de méthodes de détection d'anomalies non supervisée pour l'auscultation des ouvrages d'art. In *RFIAP*, 2024.

[3] D. Gudovskiy, S. Ishizaka, and K. Kozuka. CFLOW-AD : Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *WACV*, 2022.

[4] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. ImageNet : A large-scale hierarchical image database. In *CVPR*, 2009.

[5] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real NVP. *arXiv 1605.08803*, 2016.

[6] V. Zavrtanik, M. Kristan, and D. Skočaj. DRAEM-a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proc. IEEE/CVF International Conf. on CV*, 2021.

[7] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graph.*, 2003.

[8] J. Tan, B. Hou, T. Day, J. Simpson, D. Rueckert, and B. Kainz. Detecting outliers with Poisson image interpolation. In *MICCAI*, 2021.

[9] X. Yao, R. Li, J. Zhang, J. Sun, and C. Zhang. Explicit boundary guided semi-push-pull contrastive learning for supervised anomaly detection. In *CVPR*, 2023.

[10] S. Wang, Y. Li, H. Luo, and C. Bi. CL-Flow : Strengthening the Normalizing Flows by Contrastive Learning for Better Anomaly Detection. *arXiv preprint*, 2023.

[11] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In *CVPR*, 2019.

[12] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Trans. on Intelligent Transportation Systems*, 2019.

[13] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang. DeepCrack : Learning Hierarchical Convolutional Features for Crack Detection. *IEEE Trans. on Image Processing*, 2018.

[14] M. Mundt, S. Majumder, S. Murali, P. Panetsos, and V. Ramesh. Meta-learning convolutional neural architectures for multi-target concrete defect classification with the COncrete DEfect BRidge IMage dataset, 2019.

[15] M. Tan and Q. Le. EfficientNet : Rethinking model scaling for convolutional neural networks. In *International Conf. on machine learning*, 2019.