Projet Image - Compte-rendu 3

Sécurité Visuelle - Obscuration d'image

Loïc Kerbaul - Valentin Noyé

4 novembre 2024

1 Introduction

Nous avions présenté dans le premier compte-rendu l'état de l'art des techniques d'obscuration et d'évaluation de l'obscuration de l'image, par méthodes classiques ou encore par IA. Le second compte-rendu nous a permis de mettre en place le cadre de ce projet, en y implémentant tout d'abord les méthodes classiques d'obscuration et d'évaluation de l'image obscurcie. Ce compte-rendu est alors consacré à des recherches plus poussées et détaillées sur ces techniques par intelligence artificielle.

2 Second état de l'art des méthodes d'obscuration par IA

2.1 Auto-encodeur

2.1.1 Structure d'un auto-encodeur

Un auto-encodeur est constitué de deux parties, l'encodage et le décodage. L'objectif est de minimiser la différence entre l'entrée et la sortie. La figure 1 tirée de Song et al. [7] montre l'architecture d'un tel auto-encodeur convolutionnel.

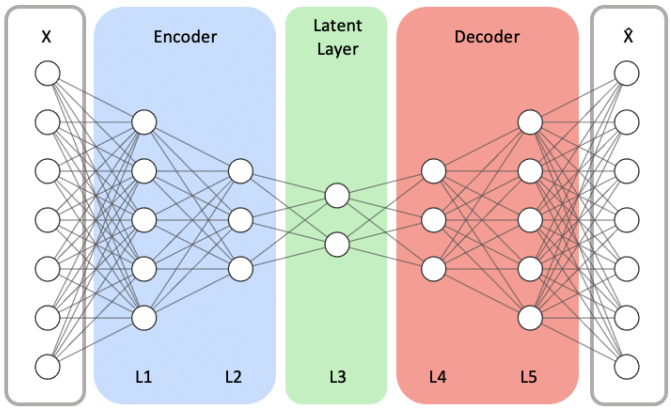

FIGURE 1 – Architecture d'un auto-encodeur

Le fonctionnement est alors le suivant :

1. Tout d'abord, un encodeur $q(x)$, prend en entrée une image x et génère une représentation latente z :

$$z = q(x)$$

2. z peut ensuite être manipulé dans l'espace latent, ou maintenu tel quel, par exemple si nous visons à compresser des données par IA.
3. Le vecteur latent modifié z' est ensuite passé dans le décodeur pour générer une nouvelle image x' :

$$x' = p(z)$$

Un auto-encodeur convolutionnel est composé de plusieurs couches de convolutions pour l'encodeur et de couches de convolution transposée pour le décodeur. L'auto-encodeur est entraîné pour minimiser une fonction de perte de reconstruction, telle que l'erreur quadratique moyenne entre l'image d'entrée x et la reconstruction x', dont nous rappelons la formule :

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N} \|x_i - x'_i\|^2$$

où N est le nombre de pixels par image. Nous notons que pour un encodage et décodage correct et sans modification de l'espace latent,

— si le MSE est trop faible, cela signifie que le modèle reconstruit parfaitement les données d'entrée, mais cela peut indiquer que le VAE se contente de mémoriser les données plutôt que de généraliser.

— si le MSE est trop élevé, cela peut indiquer que le VAE ne reconstruit pas bien les données.

2.1.2 Manipulation de l'espace latent

Une fois les images encodées dans l'espace latent, certaines caractéristiques de z peuvent être manipulées pour altérer l'image de manière contrôlée. Par exemple, une méthode simple est de mettre à zéro certaines données du vecteur z, ce qui peut supprimer certaines informations de l'image originale.

Soit $z = [z_1, z_2, \ldots, z_n]$, avec n la taille de l'espace latent. Nous créons un vecteur modifié z' en fixant certaines caractéristiques à zéro :

$$z'_i = \begin{cases} z_i & \text{si } i \notin S \\ 0 & \text{si } i \in S \end{cases}$$

où S est un sous-ensemble d'indices représentant les caractéristiques latentes que nous souhaitons mettre à zéro. Or, nous tombons sur un problème avec cette méthode, puisque nous ne pouvons pas directement interpréter l'espace latent et les caractéristiques associées à chaque valeur latente, il n'est pas toujours clair de ce que l'on va modifier. Cela peut nécessiter une analyse préalable pour comprendre quelles dimensions influencent quelles caractéristiques visuelles.

La modification de certaines caractéristiques dans z entraîne alors une génération d'images qui diffèrent des images originales par la perte ou la modification des caractéristiques associées aux variables latentes modifiées ou masquées.

2.1.3 Auto-encodeurs variationnels (VAE)

Nous notons qu'il existe également une amélioration des auto-encodeurs qui vise à introduire un concept probabiliste dans leur architecture. Ces VAE produit des paramètres d'une distribution normale (la moyenne et l'écart-type) pour chaque point d'entrée, au lieu de produire directement un vecteur de caractéristiques fixe, et permet au décodeur de générer des données à partir de cette distribution latente.

Au lieu de minimiser uniquement la différence entre l'entrée et la sortie, les VAE ajoutent une contrainte : la distribution latente doit suivre une distribution normale standard, et utilise une fonction de perte en plus telle que la divergence de Kullback-Leibler.

2.2 Réseaux de neurones génératifs antagonistes (GAN)

2.2.1 Structure d'un GAN

Un GAN se compose de deux réseaux de neurones :

— Le générateur dont le rôle est de créer des données synthétiques à partir d'un bruit aléatoire. Il tente de produire des données qui ressemblent le plus possible aux données réelles.

— Le discriminateur qui a pour tâche de distinguer les données réelles des données générées. Il prend en entrée des données réelles et des données générées et essaie de déterminer si chaque donnée provient du générateur ou d'une véritable source.

La figure 2 montre l'architecture d'un tel réseau, tiré de l'étude des GANs menée par Little et al. [6]

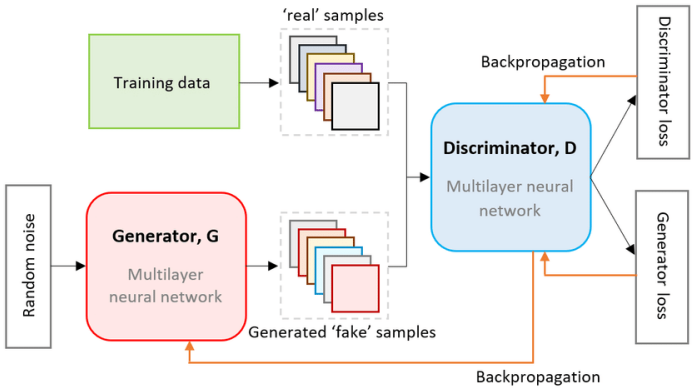

FIGURE 2 – Architecture d'un GAN

2.2.2 Entraînement

L'entraînement fonctionne d'abord avec le générateur créeant des données à partir d'un vecteur de bruit aléatoire. Le discriminateur évalue ensuite les données en recevant à la fois des données réelles et des données générées par le générateur. Il essaie alors de classifier ces données comme réelles ou générées.

Ensuite, les réseaux sont mis à jour par backpropagation, d'abord le discriminateur est mis à jour pour mieux différencier les vraies données des fausses, et générateur est mis à jour pour produire de meilleures données, dans le but de tromper le discriminateur.

Ces deux réseaux tentent alors de minimiser leur fonction de perte, et celles-ci sont classiques pour des GANs, où D et G représentent le discriminateur et le générateur respectivement :

— Le discriminateur essaie de minimiser son erreur de classification entre les images synthétiques et réelles par

$$L_D = -E(\log(D(x))) - E(\log(1 - D(G(x))))$$

où E représente l'espérance.

— Le générateur essaie de minimiser sa capacité à être reconnue en tant qu'image synthétique par le discriminateur

$$L_G = -E(\log D(G(x)))$$

Il existe d'autres fonctions de pertes permettant d'obtenir des résultats différents, tels que la perte par distance de Wasserstein (WGAN), Wasserstein avec pénalité de gradient (WGAN-GP), LSGAN (Least Squares GAN), etc.

2.3 Remplissage

2.3.1 Remplissage par GAN

Basé sur notre étude dans les précédentes sections, nous pouvons alors employer une méthode d'offuscation profonde d'un visage par remplacement avec StyleGAN [5] :

1. Pour extraire les caractéristiques faciales, un réseau comme ArcFace introduit par Deng et al. [2], qui est efficace pour les correspondances d'identité, est utilisé pour obtenir un vecteur caractéristique unique.
2. La projection de l'image d'entrée dans l'espace latent de StyleGAN est réalisée en optimisant z afin de minimiser la distance entre l'image générée et l'image originale.
3. Ensuite, afin de modifier suffisamment le visage, on ajoute une perturbation gaussienne où sa variance contrôle l'intensité du changement de caractéristiques.
4. La génération de la nouvelle image est réalisée par le générateur de StyleGAN en utilisant le vecteur latent perturbé. La fusion du visage généré avec l'image originale est ensuite appliquée.

Cela nécessite peut être certains ajustements ou post-traitements afin d'obtenir de meilleurs résultats.

2.3.2 Inpainting par convolutions dilatées

L'article par Yang et al. [8] met par exemple en oeuvre l'idée d'un remplacement par réseau de convolutions dilatées. Ces convolutions dilatées introduisent des espaces entre les éléments du noyau, comme nous le voyons dans la figure 3 tirée d'une étude menée par Cui et al. [1]. Cela permet au filtre convolutif de couvrir un champ réceptif plus large sans augmenter le nombre de paramètres ou la quantité de calcul, et donc de capturer plus de contexte.

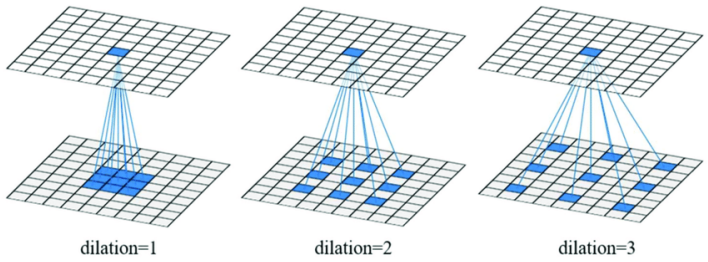

FIGURE 3 – Convolution dilatée

3 Second état de l'art des méthodes d'évaluation de l'obscuration par IA

3.1 Fréchet Inception Distance (FID)

La métrique FID [4] est utilisée pour évaluer la qualité des images générées par des modèles d'apprentissage automatique, notamment les GANs. Elle permet de quantifier la distance entre la distribution des caractéristiques des images générées et celle des images réelles. Elle repose

sur un modèle de reconnaissance d'images pré-entraîné, en l'occurrence l'Inception v3, et est calculée de la sorte :

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2\sqrt{\Sigma_r \Sigma_g})$$

avec :

— μ : la moyenne des caractéristiques extraites des images réelles et générées.
— Σ : la matrice de covariance des caractéristiques extraites des images réelles et générées.
— Tr : la trace d'une matrice, qui est la somme des éléments diagonaux de la matrice.

On s'attend ainsi à obtenir des valeurs FID faibles, et zéro indique un modèle parfait.

3.2 Classification d'images réelles et synthétiques

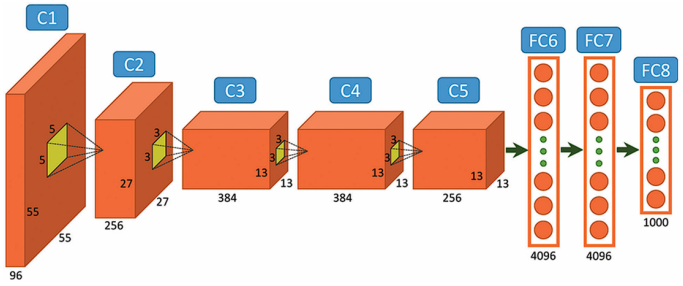

FIGURE 4 – Architecture du réseau convolutif AlexNet

Lorsque nous souhaitons évaluer si une image est synthétique ou non, il est préférable de recourir à des méthodes de classification par Deep Learning robustes. Le principle est simple : employer des classifieurs très prisés tels qu'AlexNet (voir figure 4) ou ResNet, soit pré-entraînés avec des images réelles et synthétiques, soit entraînés par nous-même sur un dataset d'images labélisées synthétiques et réelles.

Pour construire notre dataset, nous pouvons prendre diverses bases de données d'images, telles que COCO ou d'autres datasets disponibles sur le site Kaggle, puis offusquer la moitié automatiquement, soit par nos méthodes classiques, soit par des méthodes IA.

3.3 Réidentification d'offuscation par méthodes classiques

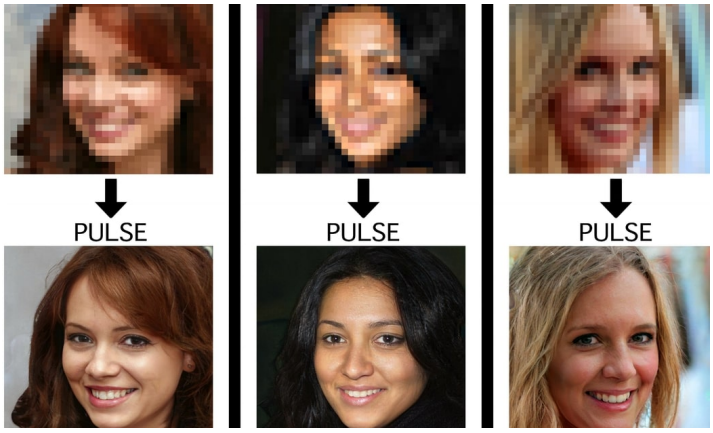

FIGURE 5 – Divers résultats obtenus par SR-GAN

Nous avions vu dans la section 2.2 qu'il était possible d'entraîner un GAN pour générer des images synthétiques plus réalistes, tel que dans la figure 5. Les SR-GAN [3] (Super Resolution GAN) sont une extension de ces réseaux GANs qui s'entraînent sur une base de données d'images ainsi que leur version sous-échantillonnée. Un tel réseau produit une sortie plus grande que son entrée, ce qui possède très souvent un grand potentiel de compression ou d'optimisation. Dans notre cas, cette méthode vise plutôt à passer outre la pixélisation de notre image et en déduire plus aisément le contenu. Les convolutions dilatées présentées dans la section 2.3.2 montre qu'il est également possible de les employer afin de remplir ces «gros pixels»intelligemment.

Nous pensons qu'il est également possible de procéder à un défloutage de cette manière, puisque l'information perdue suite à un floutage puis un sous-échantillonnage peut se rapprocher d'un sous-échantillonnage classique, mais cela dépend souvent de la taille et du type de noyau que nous avons utilisé.

Références

[1] Ximin Cui, Ke Zheng, Lianru Gao, Bing Zhang, Dong Yang, and Jinchang Ren. Multiscale spatial-spectral convolutional network with image-based framework for hyperspectral imagery classification. *Remote Sensing*, 11(19) :2220, 2019.

[2] Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, and Stefanos Zafeiriou. Arcface : Additive angular margin loss for deep face recognition. *arXiv preprint arXiv :1801.07698*, 2018.

[3] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2) :295–307, 2016.

[4] Håkon Hukkelås and Frank Lindseth. Deepprivacy2 : Towards realistic full-body anonymization. *arXiv preprint arXiv :2211.09454*, 2022.

[5] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv :1812.04948*, 2018.

[6] Claire Little, Mark James Elliot, Richard Allmendinger, and Sahel Shariati Samani. Generative adversarial networks for synthetic data generation : A comparative study. *arXiv preprint arXiv :2112.01925*, 2021.

[7] Youngrok Song, Sangwon Hyun, and Yun-Gyung Cheong. Analysis of autoencoders for network intrusion detection. *Sensors*, 21(13) :4294, 2021.

[8] Qiangpeng Yang, Hongsheng Jin, Jun Huang, and Wei Lin. Swaptext : Image based texts transfer in scenes. *arXiv preprint arXiv :2003.08152*, 2020.