

RÉALISER UN POSTER SCIENTIFIQUE

Projet Image – Master Imagine



Nicolas Dibot - 2023

Des exemples des années précédentes

Restauration de photographies anciennes

Introduction

Nous proposons de restaurer les anciennes photos qui souffrent de dégradations grâce à deux approches : avec et sans apprentissage profond. La première méthode consiste à traiter séparément les dégradations particulières aux anciennes photos et l'approche par réseau neuronal utilise une série de réseaux, tous basés sur l'article de Wan Z, Zhang B, Chen D, Zhang Y, Chen D, Wen F, Luo J. Old Photo Restoration via Deep Latent Space Translation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022).

Les méthodologies proposées ci-dessous démontrent des performances de qualité de différents niveaux en fonction de la complexité des approches.

Types de dégradations

Les vieilles photos souffrent d'habitude de deux type de dégradation :

Les dégradations structurées :

rayures et taches



Les dégradations non structurées :

bruit, flou et décoloration



Méthodes

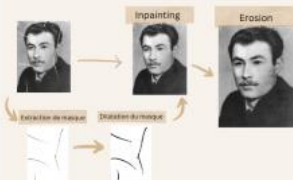
Sans apprentissage profond

Pour résoudre un problème de dégradation structurée, nous avons utilisé la méthode inpainting :

- Cette méthode d'inpainting consiste donc à trouver les pixels d'une image qui ont subi une dégradation, en utilisant un intervalle de couleurs claires comme indicateur. Ces pixels sont ensuite remplacés par les pixels les plus proches de la région non dégradée de l'image, afin de restaurer l'image à son apparence originale.
- Ensuite, pour avoir des résultats plus précis, nous avons fait une dilataion du masque, ce qui améliore la correction par rapport au le voisinage.

Par contre, il faut corriger les bruits non structurés :

- Pour cela, nous avons utilisé la méthode d'érosion, qui consiste à récupérer les pixels de la partie dégradée qui ont au moins un voisinage avec la partie propre, et les données le couleur de ce pixel voisin. Un noyau effectue cette récupération en passant sur les pixels qui ont les valeurs les plus élevées, et vérifie s'ils ont un voisinage de la partie non dégradée.



Avec apprentissage profond

On a traité le problème de restauration des images anciennes comme un problème de traduction d'images, c'est-à-dire, on souhaite apprendre le mappage entre les images propres et les anciennes photos et passer de l'un vers l'autre. Pour faire une traduction d'image on a défini 3 domaines :

- R : images réelles
- X : images artificiellement dégradées
- Y : image propre des mêmes images de X mais sans dégradation

Puis, on a effectué la traduction, au niveau des espaces latents de chaque domaine. Alors, on a utilisé le VAE (Variational Autoencoder) qui prend comme entrée une base de données, et la compresse convolutionnellement par un encodeur pour avoir un vecteur représentatif qu'on appelle espace latent, et après il l'utilise pour générer des nouvelles images, par la suite aux images d'entrée, par un decodeur.

Deux VAEs sont utilisés, un pour les images des domaines R et X , et un autre pour Y . Comme les deux domaines X et R ont chacun sa propre base de données, il faut que le VAE correspondant à ces deux domaines produise un espace latent pour les deux. Pour ce but on a utilisé un discriminateur contradictoire (adversarial discriminator), qui est, d'habitude, un réseau utilisé dans le GAN.

Qu'est ce qu'un GAN?

Le GAN, ou Generative Adversarial Network, est une technique de Machine Learning qui se repose sur la mise en compétition de deux réseaux "générateur" et "discriminateur" au sein d'un framework.

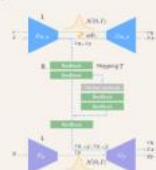


Après réalisation des deux VAEs, et du réseau de mappage, le trajet de prédiction sera :

- Encoder par le premier VAE «
- Passer l'espace latent résultant par le réseau de mappage «
- Décoder par le decodeur du 2ème VAE «

Le réseau de restauration latent nécessite d'être amélioré en incorporant une branche globale qui consiste en un bloc non local qui prend en compte le contexte global. Le bloc non local utilise explicitement un masque en entrée de sorte que les pixels de la région corrompue ne soient pas adaptés pour compléter ces zones.

La branche globale est spécifiquement destinée à l'inpainting et est fusionnée avec la branche locale sous la direction du masque. De cette façon, les deux branches constituent le réseau de restauration latent, qui est capable de traiter les dégradations diverses des photos anciennes.



Résultats

Pour évaluer nos résultats, nous avons utilisé deux outils de mesure : le rapport signal/bruit de pointe (PSNR) et la similarité de l'indice structurel (SSIM).

La comparaison est menée entre la photo originale non-dégradée du domaine Y et l'image résultante de la méthode de restauration spécifique. Ci-joint, nous affichons la photo artificiellement dégradée du domaine X et la version restaurée.

Pour l'approche avec apprentissage profond, nous avons implémenté une version simplifiée de la méthode expliquée.

Sans apprentissage profond



Avec apprentissage profond



Conclusion

Au vu des résultats, nous constatons que notre restauration sans apprentissage profond se comporte très bien lorsqu'il s'agit de reconstruire des parties plus uniformes. Un point faible est lorsqu'une rayure se trouve sur un détail ainsi qu'en raison de l'approche de traitement des défauts non structurés, le contraste et la luminosité de la photo changent légèrement.

L'approche d'apprentissage profond implémentée propose également des résultats performants. Cependant, il y a encore beaucoup de travail à mettre en œuvre pour pouvoir avoir des résultats de qualité comparable à ceux proposés dans la méthode avec apprentissage profond.

qu'en pensez-vous ?

Points forts

Points faibles

Des exemples des années précédentes

qu'en pensez-vous ?

Points forts

Points faibles

Cartoonisation

Méthode traditionnelle de traitement d'images

Image d'exemple

Cette méthode se décompose en 4 sous-parties décrites ci-après :

- **Filtre bilatéral** : Permet de réduire le bruit de l'image tout en préservant les contours de l'image.
- **Détection de contours** : Cette étape nous permet d'extraire les contours de notre image, caractéristique très présente dans les cartoons.
- **Quantification de couleurs avec k-Mean** : Permet de quantifier notre image avec un nombre de couleurs maximal et de regrouper, selon leurs similarités, les couleurs de celle-ci afin d'avoir une palette de couleurs restreinte.
- **Cartoonisation** : On obtient l'image cartoonisée en fusionnant les 2 images précédentes.

1. Filtre bilatéral **2. Détection de contours**

3. Quantification de couleurs **4. Cartoonisation**

Méthode Deep Learning avec White Box

L'image d'entrée est représentée de trois façons :

- **Représentation de surface** : En utilisant un filtrage guidé pour le flou.
- **Représentation de structure** : Quantification en utilisant l'algorithme superpixel.
- **Représentation de texture** : Détection de contours et de détails avec un changement de couleur aléatoire.

Un modèle **GAN** est défini par :

- Un **Générateur G** qui génère de nouveaux échantillons de données similaires aux images cartoon.
- **Deux Discriminateurs** :
 - Discriminateur surface (Ds) : Compare les caractéristiques de surface extraites des sorties du générateur et les images cartoons du jeu de données.
 - Discriminateur texture (Dt) : Compare les caractéristiques de texture extraites des sorties du générateur et les images cartoons du jeu de données.

Entraînement du modèle

Image d'entrée

Représentation surface → Discriminateur surface (Ds)

Représentation structure → Générateur (G)

Représentation texture → Discriminateur texture (Dt)

Image cartoonisée obtenue à l'aide de la méthode Deep Learning

Interface graphique

Image Cartoonizer

• **Interface Web** : Nous avons développé une interface web afin de rendre notre application accessible pour le plus grand nombre depuis n'importe quel appareil.

• **Utilisation** : L'utilisateur peut choisir l'image qu'il souhaite cartooniser, le nombre de couleurs et l'épaisseur des contours pour la méthode traditionnelle de traitement d'images.

• **Flask** : Cette librairie python nous permet de connecter nos scripts de cartoonisation à notre interface web.

Protocole d'évaluation

À partir de 3 images utilisées dans notre projet (un visage, une photo d'animal et un paysage), nous avons demandé à un échantillon de personnes, de choisir la version cartoonisée préférée entre les deux méthodes puis avec l'originale :

Étape 1 : Choix entre les 2 méthodes

Étape 2 : Choix de l'équipe 1 face à l'originale

Résultats :


- 61.1% de préférence pour la méthode de cartoonisation par Deep Learning.
- 1 personne sur 4 a privilégié l'image cartoonisée à l'originale.

Des exemples des années précédentes

qu'en pensez-vous ?

Points forts

Points faibles



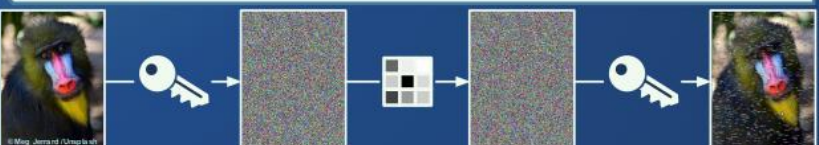
Débruitage par CNN

d'images chiffrées ou secrètes bruitées

Par [redacted]
Sous la direction de
William PUECH - Bianca JANSEN van RENSSELING - Nicolas DIBOT - Pauline PUTEAUX

Problématique

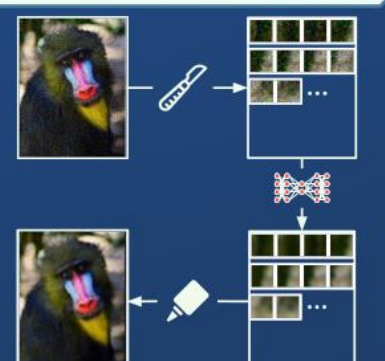
De nos jours, de plus en plus de données sont transmises sur le WEB, ou sauvegardées sur des serveurs comme les CLOUDS par exemple. Il devient alors nécessaire de se poser la question de la **sécurité** de telles communications ou de telles conservations d'informations. Une manière d'y répondre et de mettre en place un **chiffrement** de ces données. Ainsi, même si une personne malveillante arrive à intercepter ces informations, elle ne peut pas reconstruire le message en clair. Cependant, il est possible que pendant la communication (e.g. : bruit de canal) ou lors du stockage de ces données (e.g. : compression avec perte des images chiffrées), ces informations chiffrées soient **altérées**, rendant compliqué le déchiffrement ensuite. C'est sur ce constat que notre projet se base pour tenter d'établir des techniques de **débruitage** sur de telles données altérées.



Ainsi, nous pouvons formaliser ce problème via le schéma au-dessus. Il ne s'agit pas simplement de débruiteur une image, mais de la débruiteur en connaissance des informations sur le chiffrement appliqué (connaissance de l'image chiffrée et de l'image déchiffrée bruitées).

Auto-encodeur approximation


Une des manières de débruiteur une image est d'utiliser un réseau de neurones convolutif particulier, un **auto-encodeur**. Il s'agit d'un réseau de neurones par apprentissage non supervisé de type génératif. Ce type de réseau prend dans notre cas une image bruitée en entrée, pour tenter d'en générer sa version la moins bruitée possible en sortie.




En connaissance de l'algorithme de chiffrement appliqué, ou en faisant certaines **suppositions** sur le bruit appliqué, il nous serait alors possible de **corriger** plus facilement ces zones détectées comme bruitées. C'est l'avantage de cette stratégie de correction de bruit, contrairement à des techniques de débruitage plus classiques comme des filtres ou l'utilisation d'auto-encodeurs, qui s'emploient dans des **cadres plus généraux** de débruitage d'images.

Shamir


Jusqu'à présent, il a été question, dans les schémas, de chiffrements **symétriques par clé secrète**. Une autre manière de sécuriser une image est de passer par un **partage de secrets**. On va ainsi partitionner l'image à chiffrer en **n** parts différentes de telle sorte que **k** (parmi n) parts soient suffisantes pour la reconstruire. Cela va se baser sur une utilisation d'un **polynôme** de degrés $(k-1)$ pour chaque pixel, où chaque part va stocker la valeur d'un point sur la courbe de ce polynôme, de telle sorte que la valeur en zéro de ce polynôme soit la valeur du pixel que l'on cherche à cacher. Cela permet une **meilleure responsabilisation** de la part des personnes partageant une part de l'image à sécuriser, mais augmente les sources potentielles de bruit.




Avec cette technique, on a pu constater que l'on arrive à reconstruire l'image d'origine (présence des bonnes **couleurs**, du **sujet**, des **formes**), mais que l'on obtient un résultat flou ; on a du mal, au moment de la reconstruction, à préserver des informations de **texture** notamment. Pour tenter de conserver cette information, on pourrait se servir des résultats du réseau de neurones afin d'obtenir une image depuis laquelle il nous deviendrait plus facile de détecter des parties bruitées de l'image déchiffrée.




Étape de chiffrement/déchiffrement




Bruit sur l'image



Passage à l'auto-encodeur

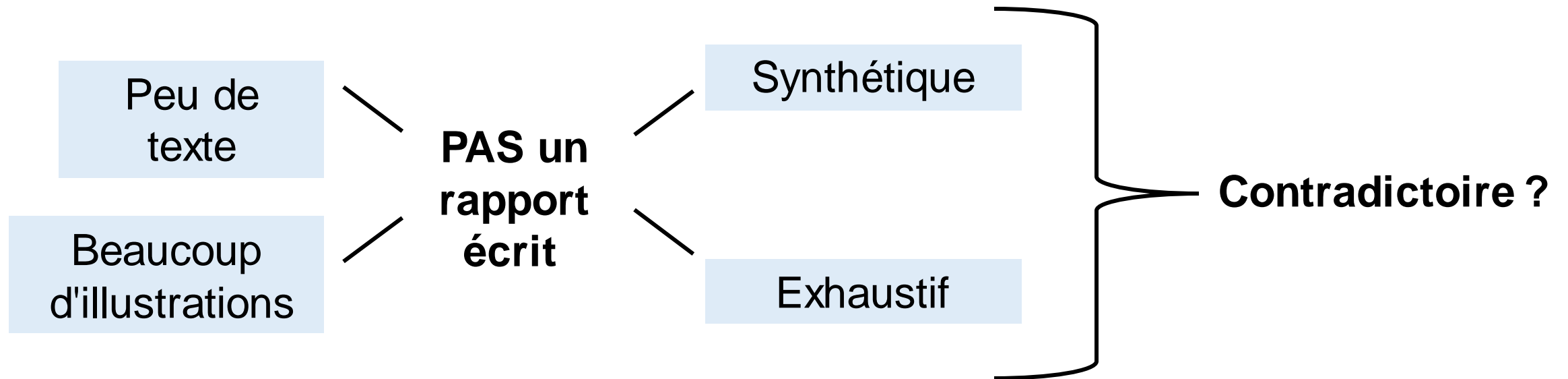


Segmentation de l'image en blocs



Assemblage des différents blocs

Le contenu



Le contenu

3 niveaux de lecture

TITRE + schéma principal

Doit pouvoir être lu de loin

**Sous-titres + illustrations
secondaires**

Doivent pouvoir être lus sans avoir les yeux collés au poster et donner une idée de la démarche générale

**Détails:
Texte, légendes, sources ...**

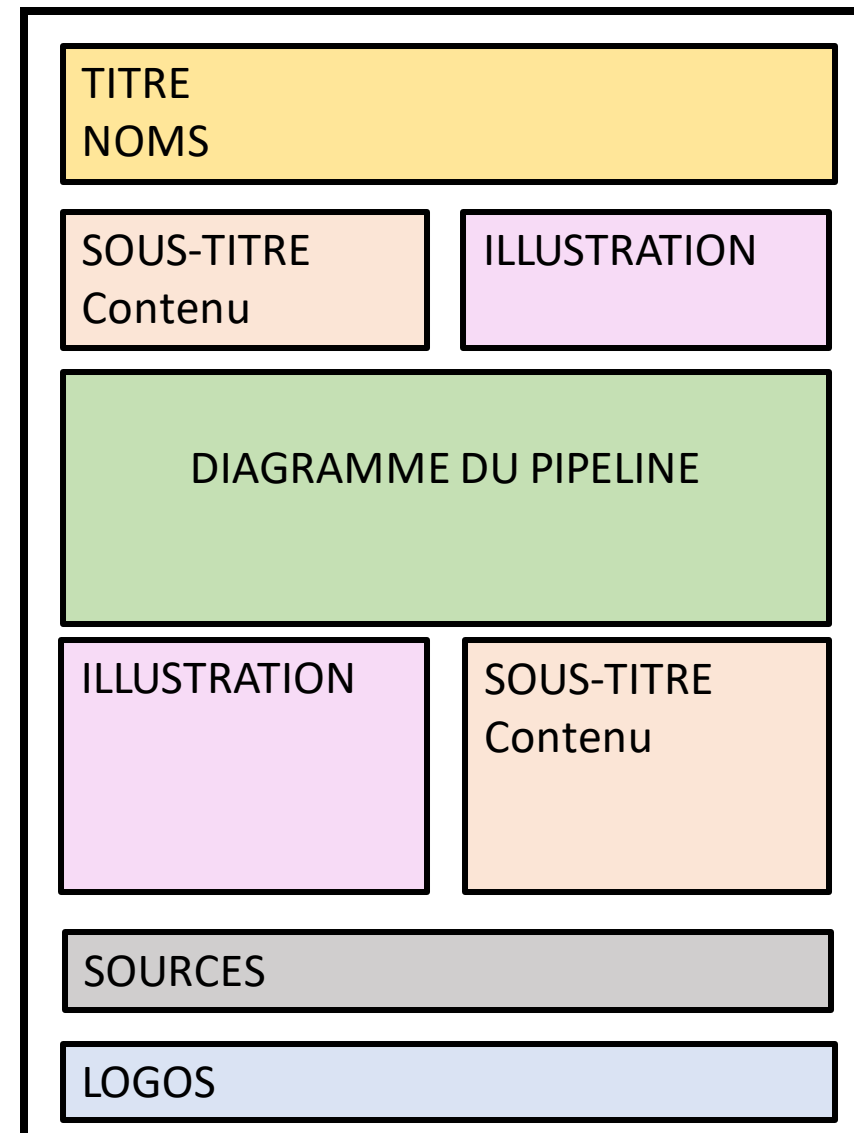
Le texte donne plus de détails, mais quand c'est possible, le remplacer par des représentations graphiques: diagrammes, plots, dessins...

Le contenu

Par où commencer ?
Une trame

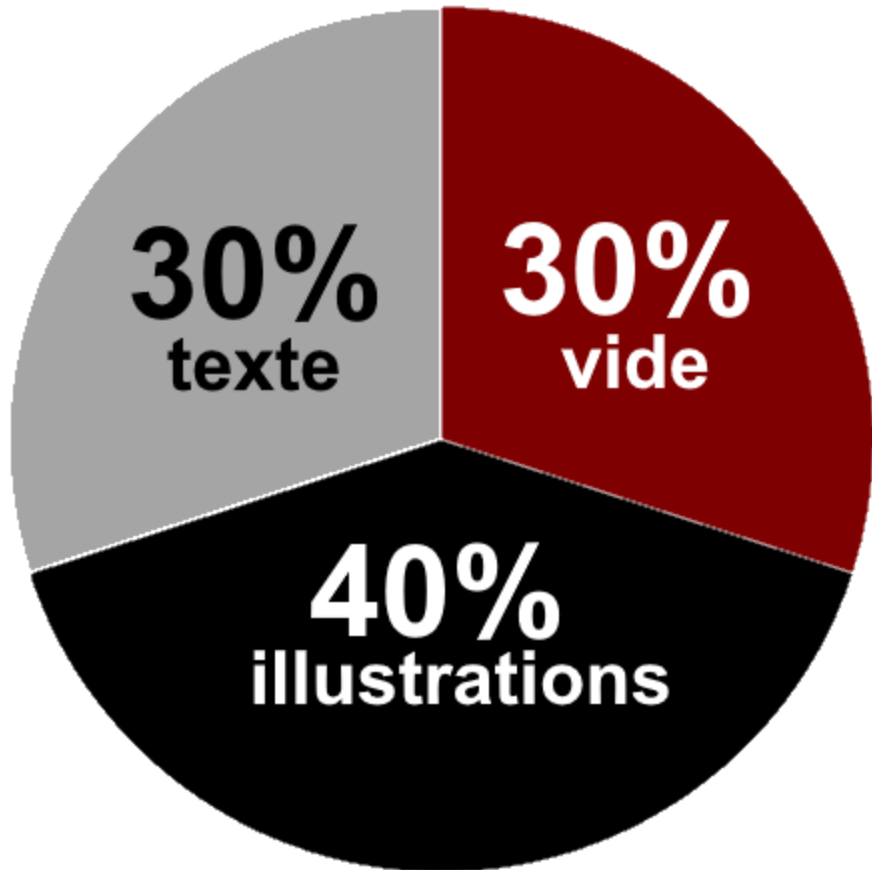


D'autres idées



Suggestion de trame d'un poster (non obligatoire)

La forme



Pas **plus de 3**
couleurs

Attention aux
contrastes pour
que ce soit lisible

Attention aux
contrastes pour
que ce soit lisible

Attention aux
contrastes pour
que ce soit lisible

Les couleurs et les formes doivent aider à comprendre la structure du poster (quelles sont les différentes parties, quelle est leur importance, leur ordre)



si besoin, zoomer sur les parties importantes des figures: attention à leur résolution/qualité