



Master 2 Imagine

Projet Image

Compte rendu N°2

Mhamad Awwad

Albena Stefanova

Restauration d'images bruitées par CNN

Table des matières :

I. Avancement global	2
II. Tâches menées sur la période écoulée	2
I.I. L'article	2
I.II. La base de code	3
III. Tâches prévues	3

I. Avancement global

Pendant cette deuxième semaine nous avons partagé le travail en deux :

- étudier plus en détails l'article '*Old Photo Restoration via Deep Latent Space Translation*' - <https://arxiv.org/pdf/2009.07047.pdf>
- définir la base de code en se basant sur le TP6 CNN - [Document Google Collaboratory](#)

Nous avons réussi à atteindre les objectifs définis la semaine dernière.

II. Tâches menées sur la période écoulée

I.I. L'article

Après avoir effectué plusieurs recherches, notre projet va être basé sur l'article "*Old Photo Restoration via Deep Latent Space Translation*". La solution de cette problématique proposée dans cet article se distingue des autres solutions qui utilisent le Deep Learning pour faire la restauration des images synthétiquement anciennes (artificiellement anciennes), et pas les anciennes images réelles. De plus, cette solution prend en considération, les différents types de défaut des images: défauts structurés, comme les bruit de film, flou et décoloration et défauts non structurés comme les rayures et taches. Contrairement aux autres algorithmes qui traduit les image sous 2 domaines, notre algorithme traduit les images sous 3 domaines:

1. Domaine de la photo réelle
2. Domaine synthétique, où les images souffrent d'une dégradation artificielle
3. Domaine "Ground Truth", qui comprend des images sans dégradation (même image du domaine 2 mais sans dégradation)

Ensuite, on a cartographié chaque domaine à sa "latent space" correspondante en utilisant le VAE (variational autoencoder: qui sert à encoder des images avec une représentation compacte). Et comme les 2 premiers domaines souffrent d'une dégradation, on a aligné leurs latent space, ce qui conduit à encoder les fonctionnalités de toutes les images corrompues et constater un type général des images dégradées.

La restauration est réalisée en utilisant les paires de données synthétiques (domaines 2 et 3), on apprend la traduction de "latent space" des images corrompues à la "latent space" de la "Ground Truth".

L'article comprend ensuite, 3 étapes qui améliore la restauration pour améliorer les résultats obtenus:

- Restauration de dégradation multiple
- Détection des régions de défauts
- Amélioration de visage

I.II. La base de code

On a repris le code de TP6 sur Google Colab -

<https://colab.research.google.com/drive/1XOI2CuFGAuuVb5-rNc32BgI172mWJmcB?usp=sharing>. On a choisi cette plateforme car elle propose le même environnement pour tous les éditeurs et on a la possibilité d'y travailler en même temps.

Le travail effectué pendant la semaine était de s'habituer à la gestion des fichiers sur Google Drive et à leur accès dans Colab et d'adapter le code pour qu'il puisse générer le dataset.

Comme première version, on a décidé de se concentrer sur le deuxième domaine présenté [ci-dessus](#) - '*Domaine synthétique, où les images souffrent d'une dégradation artificielle*'.

D'une base de donnée très simple on génère des images de synthèse de qualité dégradée d'après différentes méthodes :

- Bruit blanc gaussien
- Bruit salt-and-pepper
- Flou (avec un filtre passe-bas)

Ensuite on a défini une méthode qui récupère les images originales et les images générées pour ensuite être utilisées comme données d'entraînement du modèle.

Tout le travail effectué ne sera pas utilisé directement la semaine prochaine. L'objectif de ce rendu est de définir une bonne base pour la partie CNN et de comprendre l'idée globale de l'article.

III. Tâches prévues

La semaine prochaine, nous avons décidé d'essayer d'implémenter une méthode de débruitage sans apprentissage profond. Elle se concentre sur les deux types principaux de défaut des images: défauts structurés, comme les bruit flou et décoloration et défauts non structurés comme les rayures et taches.

- Le premier principe est d'appliquer un filtre homogène en utilisant les pixels voisins dans un bloc local.
- Le deuxième est à échelle plus grande où il faut utiliser une méthode d'inpainting pour assurer la cohérence structurelle de l'image.