



Master 2 Imagine

Projet Image

## Compte rendu N°4

**Mhamad Awwad**

**Albena Stefanova**

---

Restauration d'images bruitées par CNN

---

## Table des matières :

I. Avancement global	<b>2</b>
II. Tâches menées sur la période écoulée	<b>2</b>
II.1. Auto-encodeur variationnel (VAE)	2
II.2. Reconstruction des images synthétiques avec CNN et génération des images dégradées	3
III. Tâches prévues	<b>5</b>

## I. Avancement global

---

Pendant cette quatrième semaine nous avons partagé le travail en trois:

- C'est quoi "auto-encodeurs variationnels" ou VAE.
- Génération manuelle des images dégradées (dégradations structurées et dégradations non-structurées).
- Reconstruction des images synthétiques avec CNN donc avec le domaine "Ground Truth" et le domaine synthétique.

Cette fois nous avons travaillé localement dans le fichier avec le joli nom de [VAEv1-pb de lecture de resultats.ipynb](#)

Nous n'avons pas réussi à atteindre les objectifs entièrement car nous n'avons pas eu du succès avec l'affichage du résultat de la prédiction de notre modèle.

## II. Tâches menées sur la période écoulée

---

### II.1. Auto-encodeur variationnel (VAE)

---

La principale force des auto-encodeurs réside dans leur capacité à extraire la représentation abstraite de l'espace de données censé gérer des instances non vues. Cela ouvre des possibilités où l'on pourrait générer de nouvelles images qui n'ont pas encore été vues en utilisant l'espace latent. L'architecture générale de l'auto-encodeur ne laisse cependant pas beaucoup de liberté dans la traversée de l'espace latent. Ceci peut être contourné par les auto-encodeurs variationnels (VAE) qui apprennent une distribution latente au lieu d'un vecteur latent et permettent donc d'interpoler dans l'espace latent. Plus précisément, un auto-encodeur variationnel modélise une distribution gaussienne multivariée qui suppose que les données peuvent être approximées comme une distribution normale.

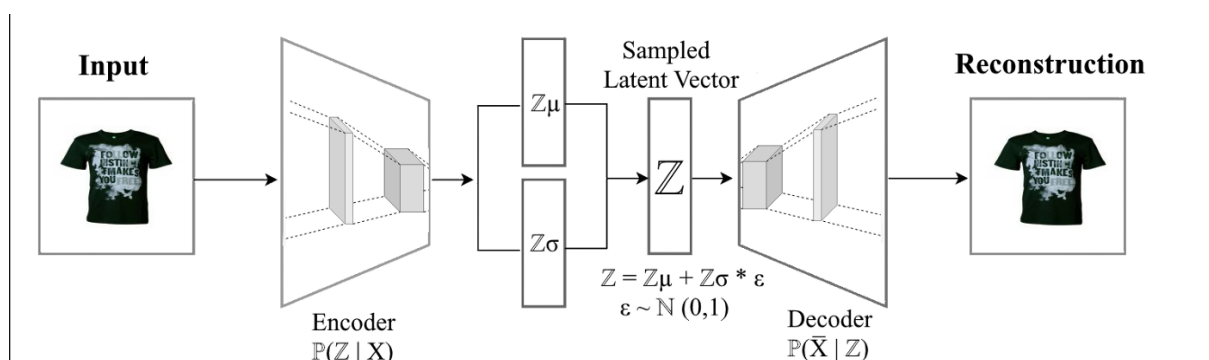


Figure 1. Modèle d'auto-encodeur variationnel (VAE)

une distribution gaussienne multivariée, qui mappe l'entrée sur l'espace latent contrairement à un seul point multidimensionnel dans l'espace latent comme dans les auto-encodeurs généraux.

Tout comme les auto-encodeurs, les VAE apprennent également en optimisant une fonction objectif qui est une fonction de perte calculée pour chaque instance de données. Cependant, la fonction de perte pour VAE est différente des auto-encodeurs en ce sens qu'ils minimisent non seulement l'erreur de reconstruction, mais appliquent également la contrainte selon laquelle le vecteur latent provient d'une distribution normale. Ceci est réalisé en ajoutant un terme supplémentaire dans la fonction de perte:

$$L(x, \hat{x}, Z_\mu, Z_\sigma) = L_{reconstruction}(x, \hat{x}) + L_{divergence}(Z_\mu, Z_\sigma)$$

L'erreur de reconstruction est calculée comme d'habitude entre l'image d'entrée et la sortie du décodeur et peut être modélisée comme une erreur quadratique moyenne:

$$L_{reconstruction}(x, \hat{x}) = 1/N \sum_{i=1}^n (x_i - \hat{x}_i)^2$$


$$L_{divergence}(Z_\mu, Z_\sigma) = L_{KL}(\mathcal{G}(Z_\mu, Z_\sigma), \mathcal{N}(0, 1))$$

L'erreur de divergence peut être calculée à l'aide de Kullback – Leibler Divergence ( KL Divergence ) qui est une mesure de la façon dont une distribution de probabilité diffère d'une autre. Puisque l'hypothèse est que les données proviennent d'une distribution normale, nous pouvons donc calculer la divergence par:

$$L_{KL}(\mathcal{G}(Z_\mu, Z_\sigma), \mathcal{N}(0, 1)) = -0.5 * \sum_{i=1}^n (1 + \log(Z_{\sigma i}^2) - Z_{\mu i}^2 - e^{\log(Z_{\sigma i}^2)})$$

## II.2. Reconstruction des images synthétiques avec CNN et génération des images dégradées

La semaine passée, on a réalisé une méthode de débruitage et de correction des images anciennes avec apprentissage profond.

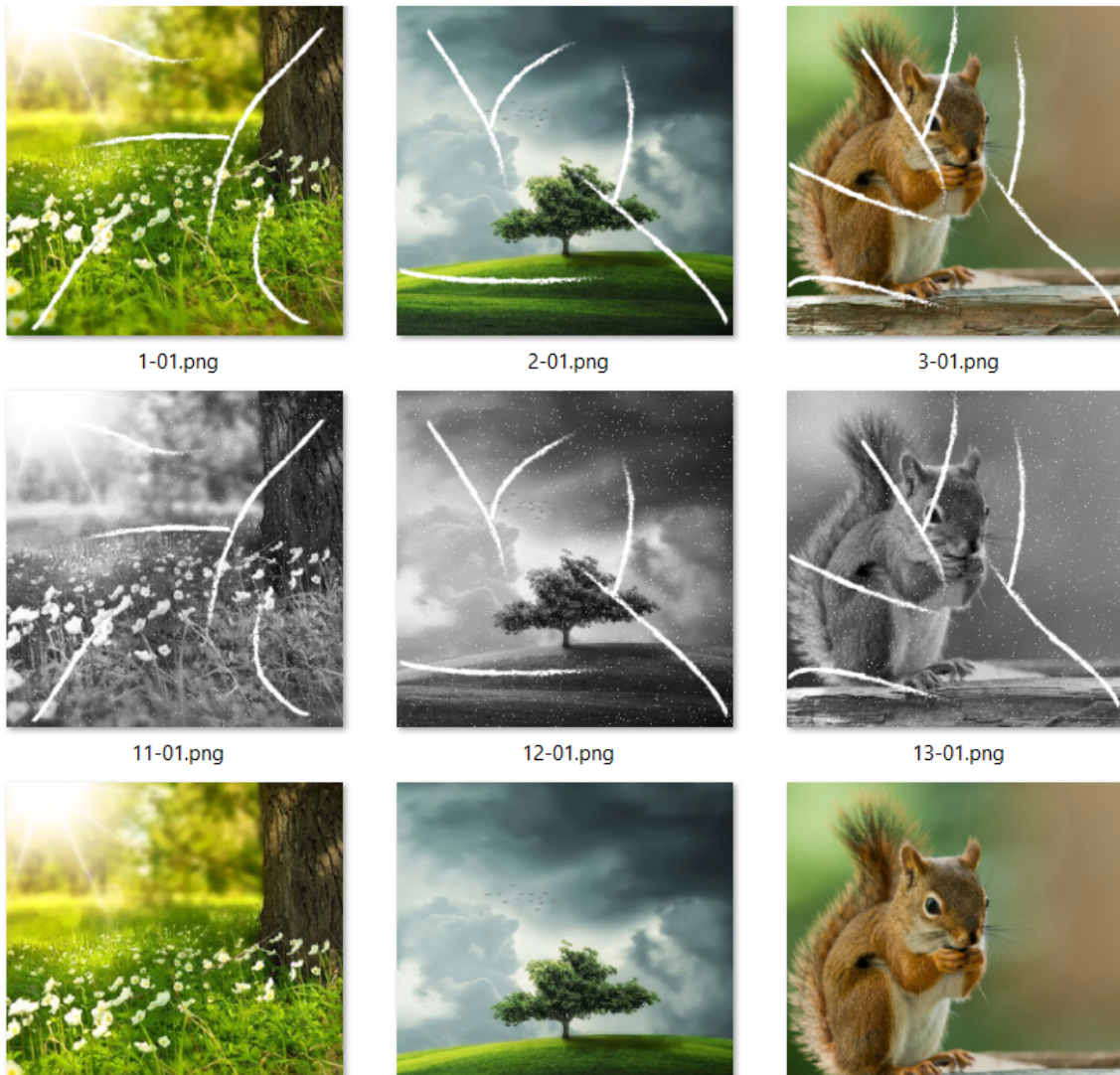
En se basant sur le Playlist  1/ Réseaux autoencodeurs (AE) , nous avons réussi à implémenter un réseau auto-encodeur de débruitage des images anciens. Comme une grande partie de travail pour cette semaine était de comprendre le fonctionnement des auto-encodeurs et auto-encodeurs variationnels, nous avons réalisé une méthode assez simpliste et très proche aux cours de M.Parouty(cours en ligne - [CNRS - FIDLE \(Formation Introduction au Deep LEarning\) - YouTube](#) ).

Le travail final se trouve dans le dossier [Jupyter](#) sur notre page GitHub.

Pour l'apprentissage de modèle, nous avons utilisé 10 images. Elles ont été dégradées de manière manuel et automatique - la dégradation structurée était effectuée sur Photoshop et la dégradation non-structurée avec un algorithme de bruitage.

Ci-dessous, on a mis 3 images et leurs trois étapes :

- dégradation structurée
- dégradation non-structurée
- image originale



*Figure 2. Première ligne - images avec des dégradations structurées.*

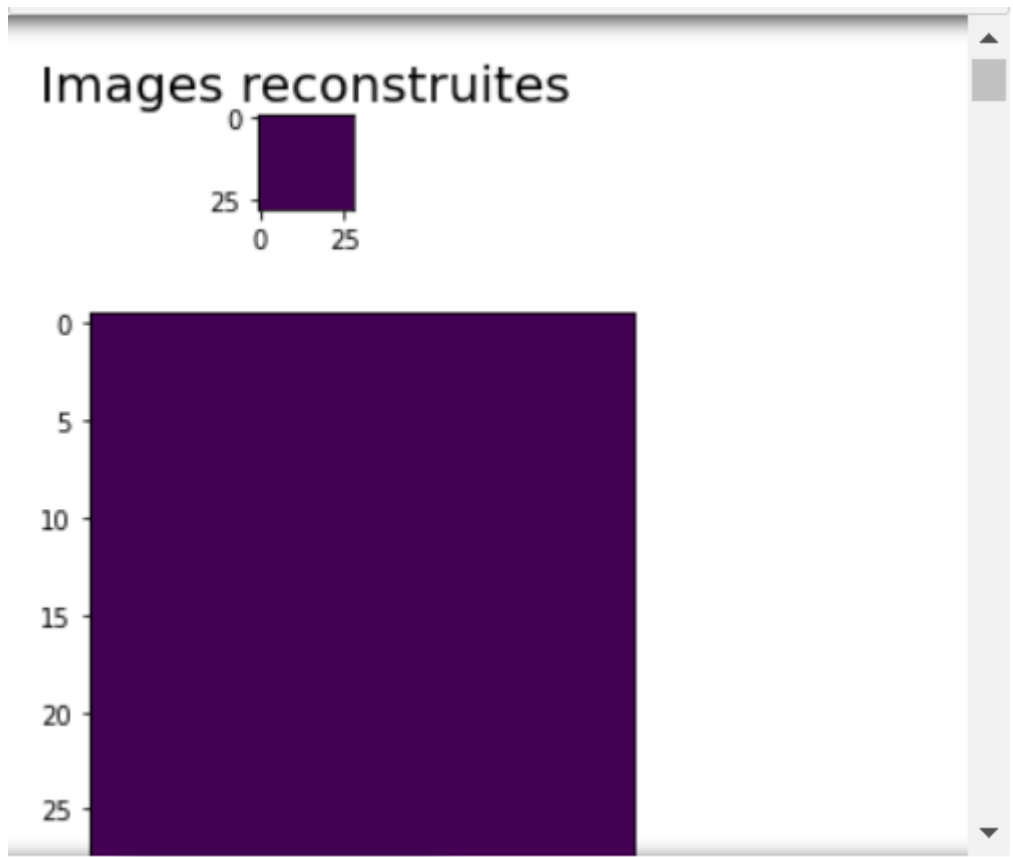
*Deuxième - images avec des dégradations non-structurées.*

*Troisième - images originales.*

Par rapport à la vraie base que nous avons envie d'utiliser, au début de la semaine nous avons envoyé un mail à un des auteurs de l'article. Ils ont implémenté leur théorie et maintenant leur projet [Old Photo Restoration \(Official PyTorch Implementation\)](#), fait partie de

Microsoft. Malheureusement, nous avons l'impression que l'auteur, Ziyu Wan, ne répondra pas.

Nous n'avons pas réussi à visualiser les prédictions de notre auto-encodeur. Ce problème nous a longtemps bloqué et nous n'avons pas avancé sur le VAE. Les résultats pour une liste des images à reconstruire en question sont ci-dessous.



*Figure 3. Affichage incorrect des résultats de la prédiction du modèle.*

### III. Tâches prévues

---

La semaine prochaine, nous avons décidé de continuer l'implémentation de la méthode de débruitage avec VAE.