



Rapport — Intelligence artificielle

CARRIER Damien — BOUSQUET Léon

SAVINAUD Clément — DUCA Arthur

Méthodologie	2
Architectures	3
Ressources	4
Rappel de formules	4
Résultats	6
Partie 1A – U-Net :	6
Partie 1B – U-Net :	9
Partie 1B – Resnet18 :	12
Partie 2 – U-Net :	15
Partie 2 – Resnet18 :	17
Conclusion	20

Méthodologie

Afin de résoudre les problématiques de restauration et de déchiffrement d'images, nous avons établi des modèles de régression. L'objectif est d'entraîner les modèles à reproduire l'image originale en partant de l'image dégradée.

Pour des raisons de capacité de nos ordinateurs personnels, nous avons pu uniquement réaliser des batchs de quatre images à la fois. Cela a grandement affecté notre capacité à tester plusieurs modèles puisque les périodes de training prenaient entre 30 minutes et 1 h par époque.

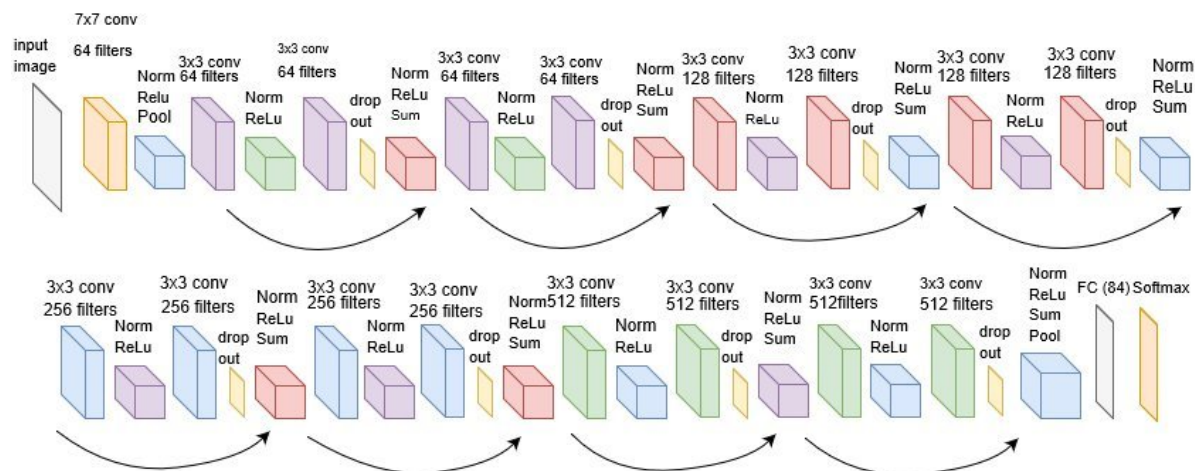
Le dataset que nous utilisons est constitué de 12 000 images d'entraînement et de 1 000 images de test. Ce qui nous donne une shape du dataset qui est de (4, 3, 96, 96).

Notre but ici, être capable de prédire une image RGB de taille 96 pixels par 96 pixels.

Architectures

Pour réaliser ce projet, nous nous sommes tout d'abord basés sur l'architecture **Resnet18** vue en cours. Nous avons pris ce modèle pour avoir une première approche sur les exercices proposés.

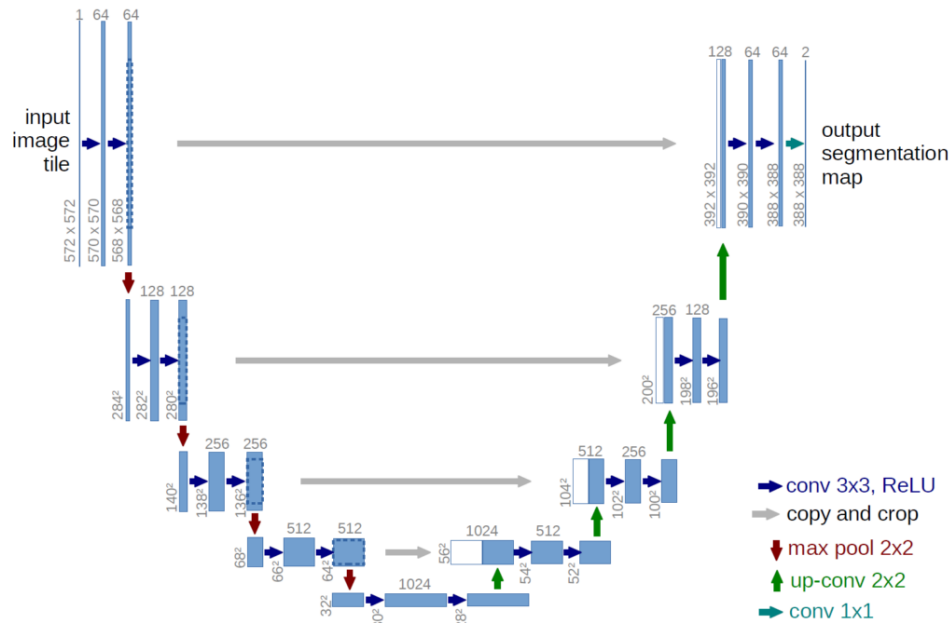
ResNet 18 est une version du Residual Network à 18 couches. Cette architecture est composée de convolutions, normalisations, pooling et de ReLU. On retrouve également des phases de dropout pendant lesquelles une partie aléatoire des neurones est désactivée pour forcer le CNN à s'adapter à un manque d'informations.



Architecture ResNet 18

Au vu de la qualité des résultats, nous avons dû penser à une autre architecture. Nous nous sommes alors intéressés à l'architecture U-Net qui a été créée pour fonctionner avec moins d'images d'entraînement et pour permettre une segmentation plus précise.

U-Net est une architecture décomposée en deux parties. La première s'appelle la **contraction**, aussi appelé **encodeur** qui est utilisé pour capturer le contexte d'une image. On y retrouve un assemblage de couches de convolution et de max pooling permettant de créer une "carte" des caractéristiques d'une image et de réduire sa taille pour diminuer le nombre de paramètres du réseau. La seconde partie est une **expansion**, aussi appelé **décodeur**. Elle permet une localisation précise.



Exemple d'architecture U-Net

Ressources

- Vous avez tous les deux (Mr Harispe et Mr Yaddaden) accès au dépôt GitLab qui se trouve à l'adresse suivante : [clik me](#)
- Les modèles déjà entraînés sont uploadé sur un cloud privé. Vous pouvez vous connecter à l'adresse suivante : [clik me](#), en utilisant les identifiants suivants. Login : clement.savinaud31@gmail.com, mot de passe : Infres2021. Si vous rencontrez des difficultés pour vous connecter, n'hésitez pas à nous contacter par mail.

Rappel de formules

```
def eval_metric(img, pred):
    return torch.abs(img - pred).sum().item()
```

Nous avons utilisé cette formule donnée dans le sujet pour évaluer nos jeux d'entraînement. Cela nous a permis d'établir le calcul d'un delta, afin de calculer le delta max, le delta min et le delta moyen.

```
# Metrique d'évaluation du model
evaluation = eval_metric(labels, y_pred)
total += evaluation
```

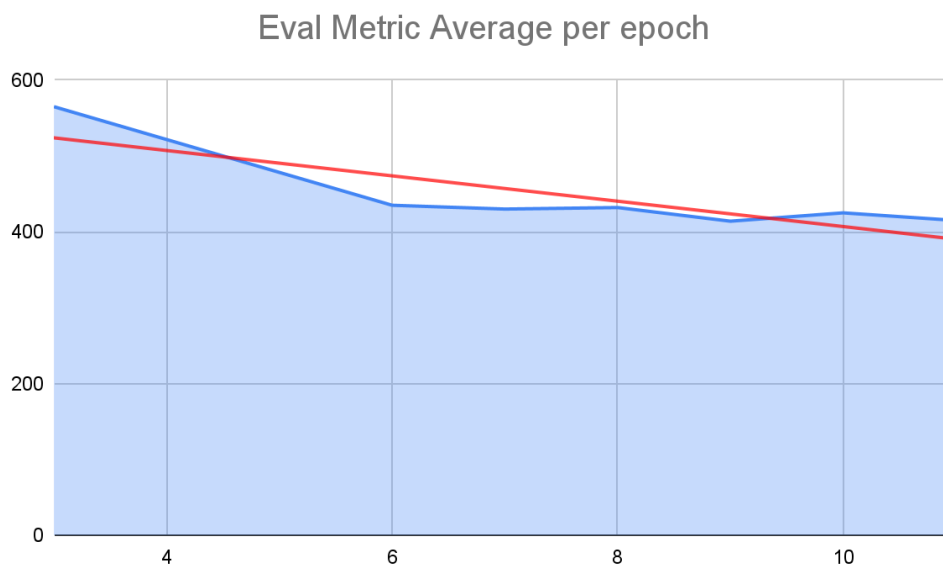
```
if (evaluation > delta_max) :  
    delta_max = evaluation  
if (evaluation < delta_min) :  
    delta_min = evaluation
```

Pour calculer ce delta, la fonction `torch.abs()` calcul pour chaque pixel de l'image prédit la valeur absolue de la différence avec l'image originale. Elle fait la somme de ces valeurs, ce qui donne le delta de différence entre les deux images.

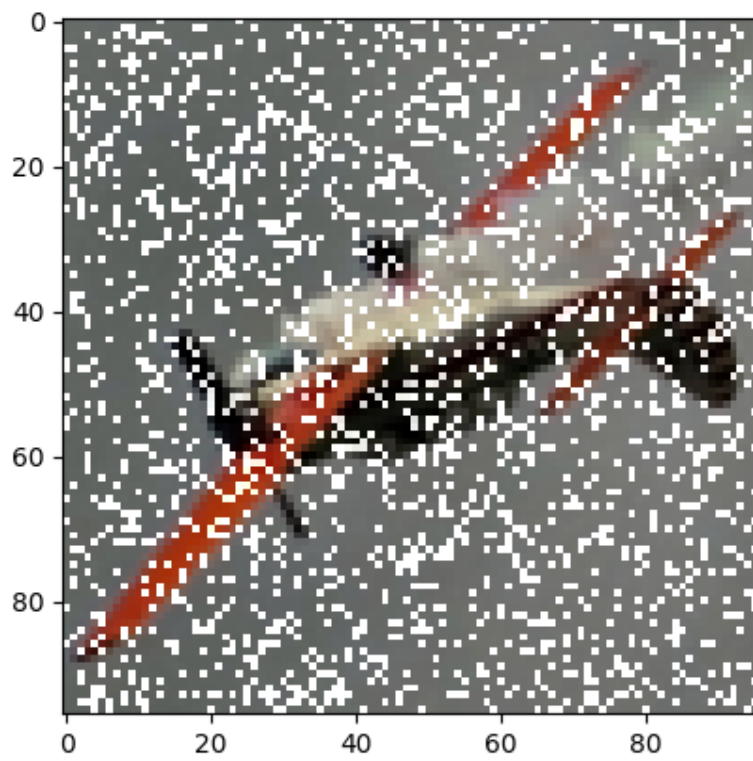
Résultats

Partie 1A – U-Net :

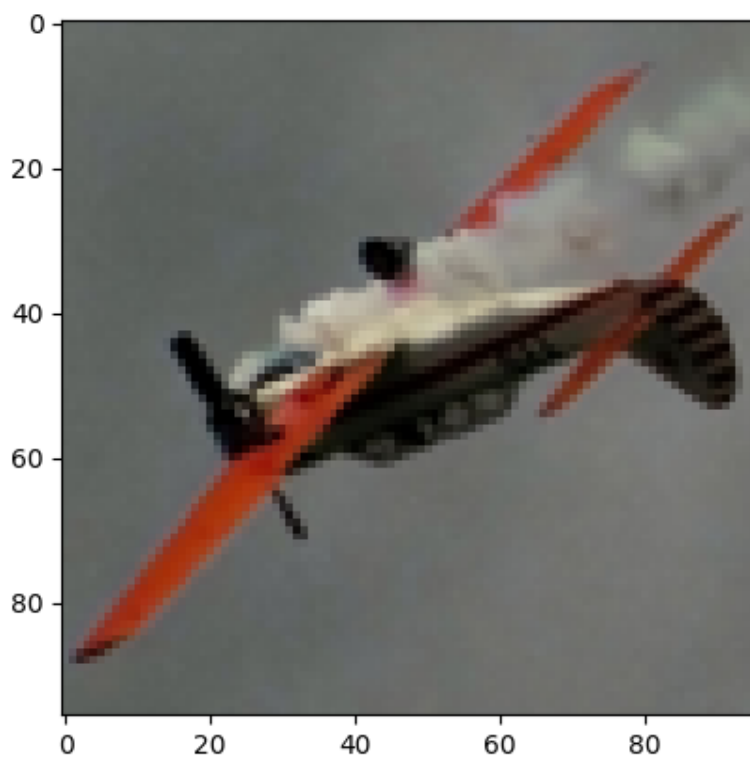
Pour cette partie nous avons réussi à obtenir un delta moyen de 413 au minimum lors de l'epoch numéro 11. Les images sont quasiment identiques, il manque encore quelques couleurs. Nous pensons que si nous pouvions lancer quelques epoch de plus, tout en vérifiant ne pas atteindre l'overfitting que nous pourrions obtenir d'encore meilleurs résultats.



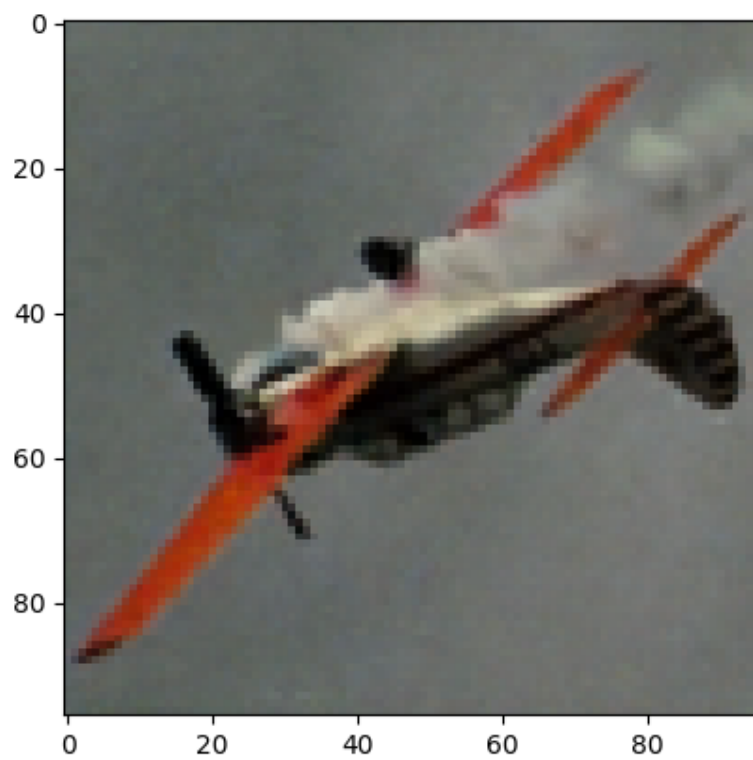
- Image de base :



- Image attendue :

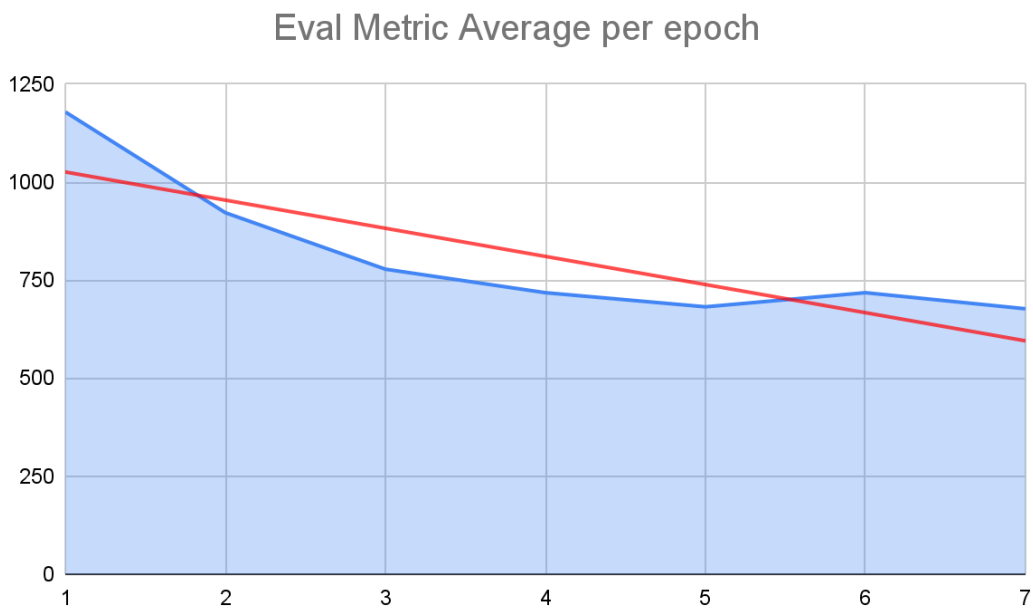


- Image prédite :



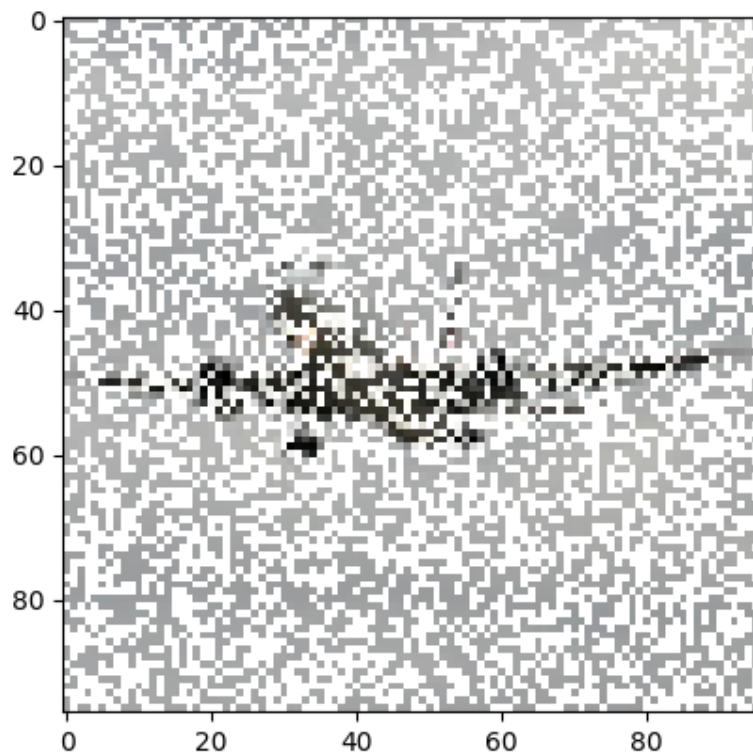
Partie 1B – U-Net :

Pour cette partie nous avons réussi à obtenir un delta moyen de 677 au minimum lors de l'epoch numéro 7. Les images sont quasiment identiques, il manque encore quelques couleurs. Nous pensons que si nous pouvions lancer quelques epoch de plus, tout en vérifiant ne pas atteindre l'overfitting que nous pourrions obtenir d'encore meilleurs résultats.

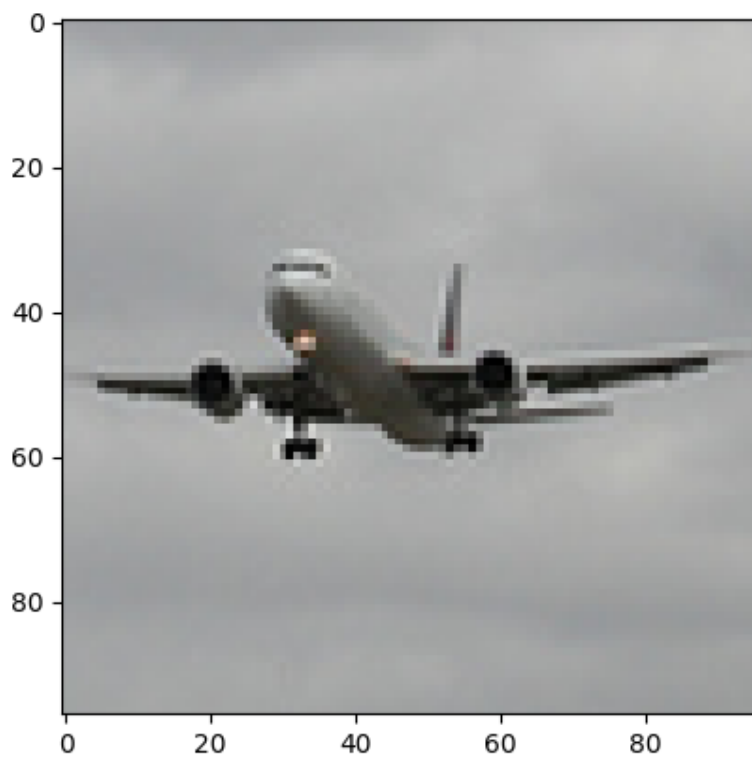


Exemple de résultats d'images après sept epochs:

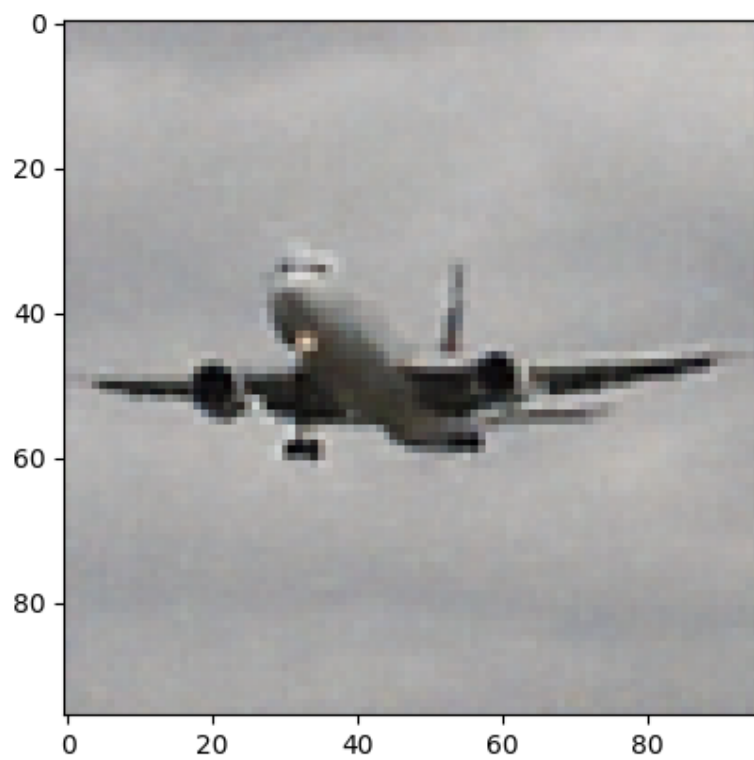
- Image de base :



- Image attendue :

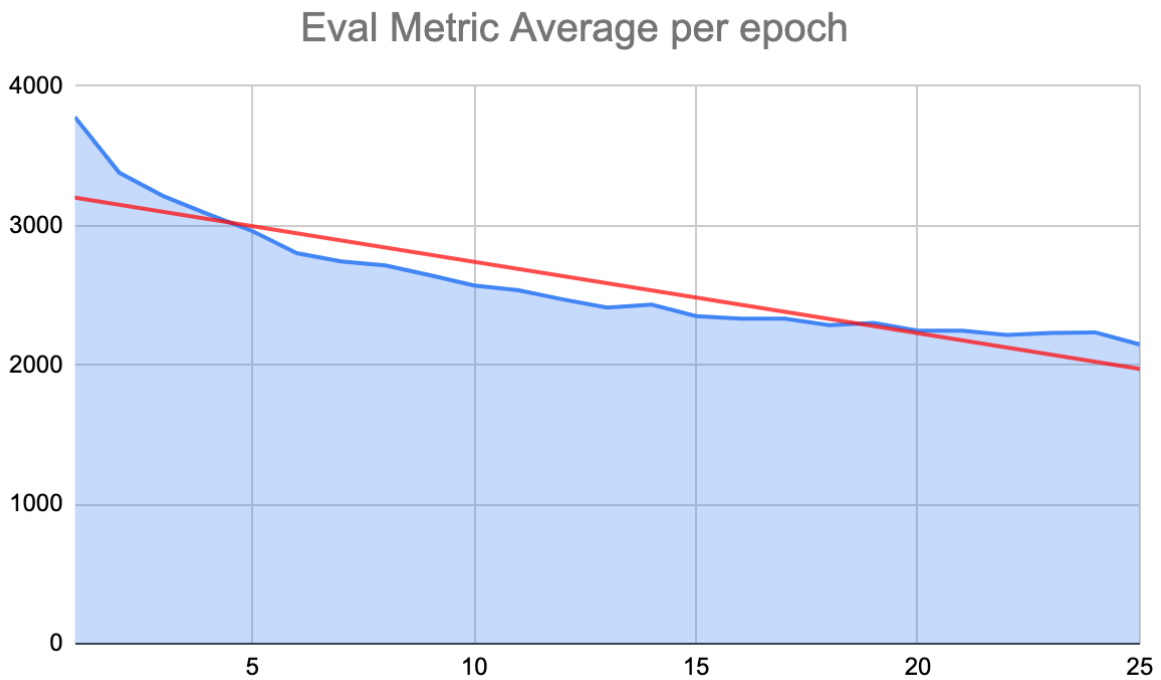


- Image prédite :

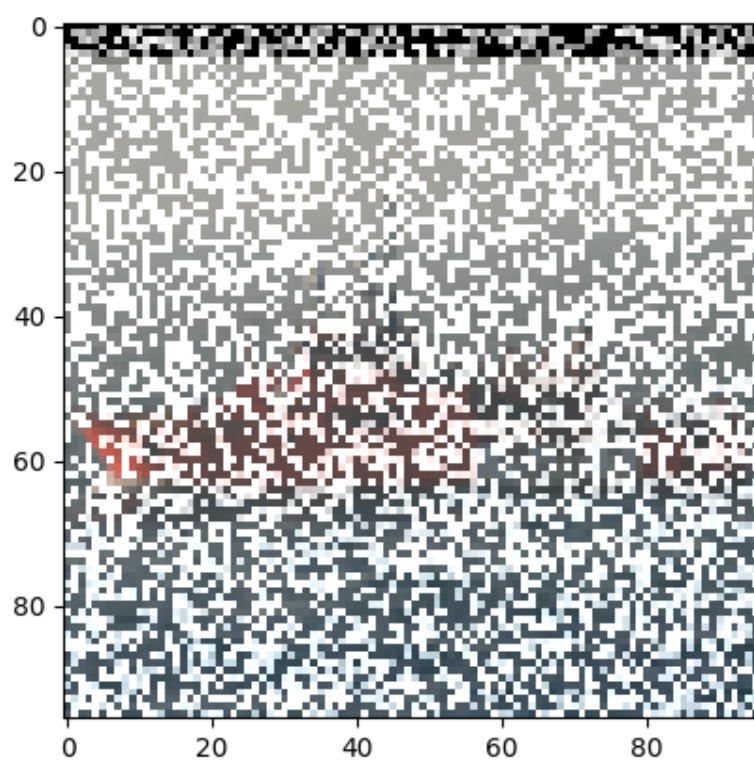


Partie 1B – Resnet18 :

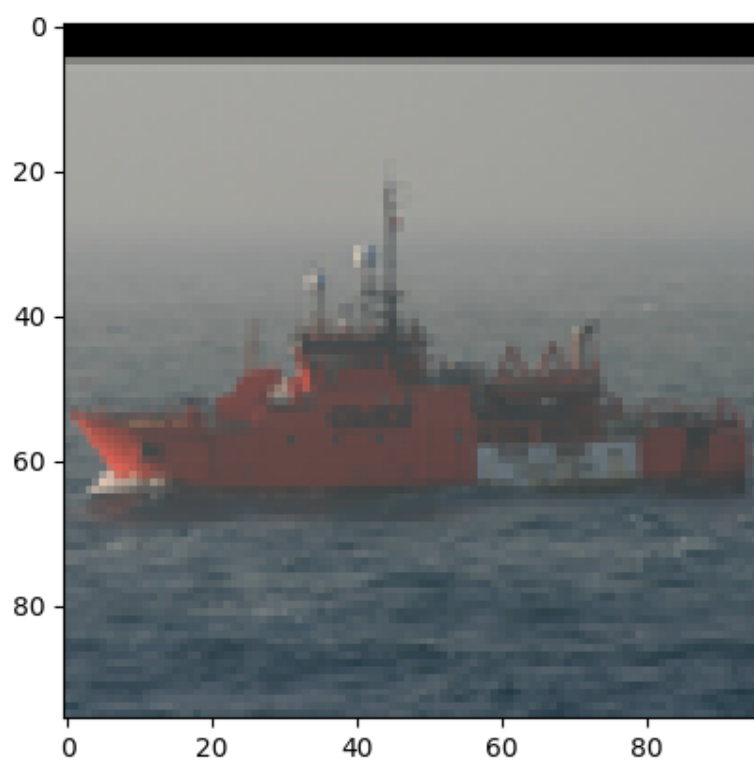
Pour cette partie nous avons réussi à obtenir un delta moyen de 2147 au minimum lors de l'epoch numéro 25. Sur l'image prédite, on peut observer la silhouette du bateau sans en distinguer les détails. Lorsque l'on regarde la courbe d'évolution de l'évaluation, on peut affirmer qu'il est encore possible d'améliorer les résultats, mais de manière très légère. Si la tendance reste la même, compte tenu du nombre d'epoch.



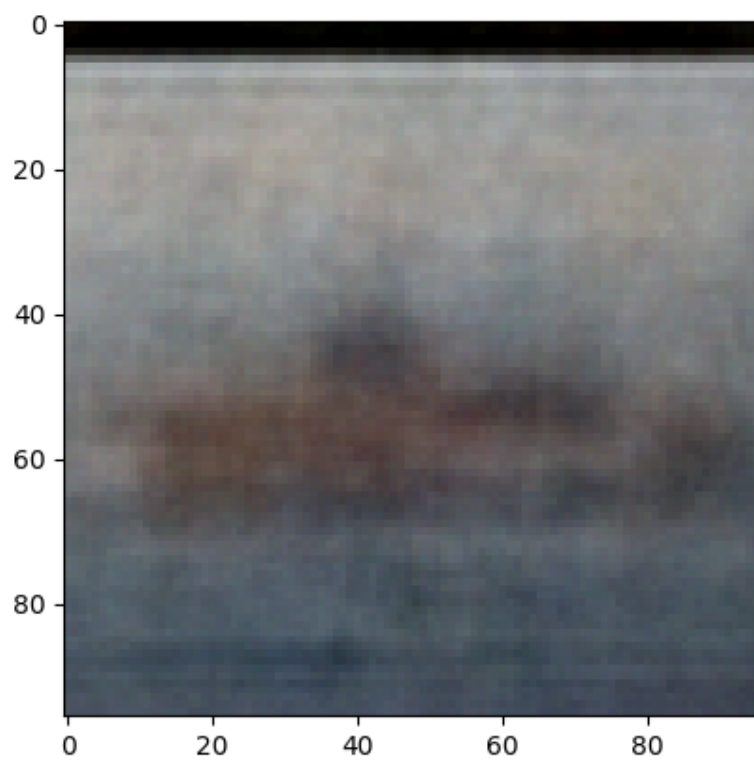
- Image de base :



- Image attendue :

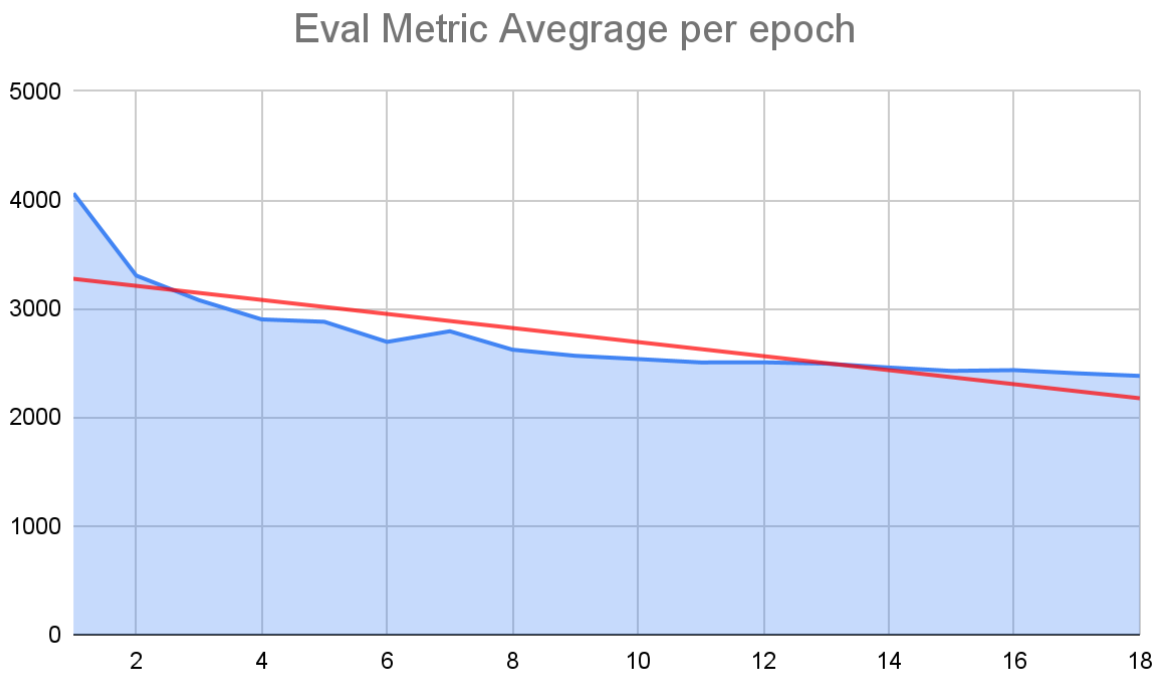


- Image prédite :



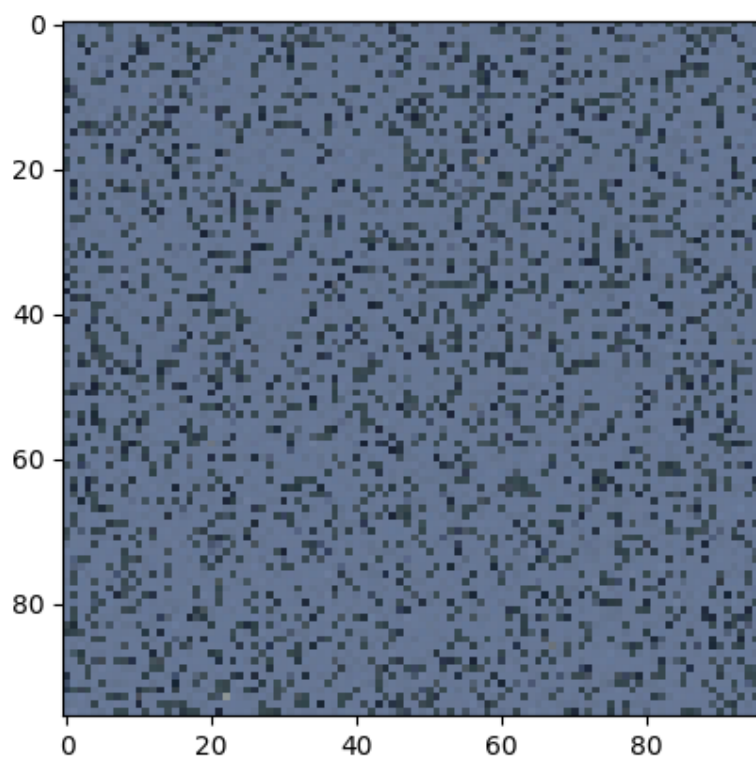
Partie 2 – U-Net :

Pour cette partie nous avons réussi à obtenir un delta moyen de 2381 au minimum lors de l'époch numéro 18. Sur l'image prédite, on peut observer la silhouette de l'avion sans en distinguer les détails. Lorsque l'on regarde la courbe d'évolution de l'évaluation, on peut affirmer qu'il est encore possible d'améliorer les résultats, mais de manière très légère. Si la tendance reste la même, on peut considérer que nous n'arriverons pas à avoir d'image détaillée.

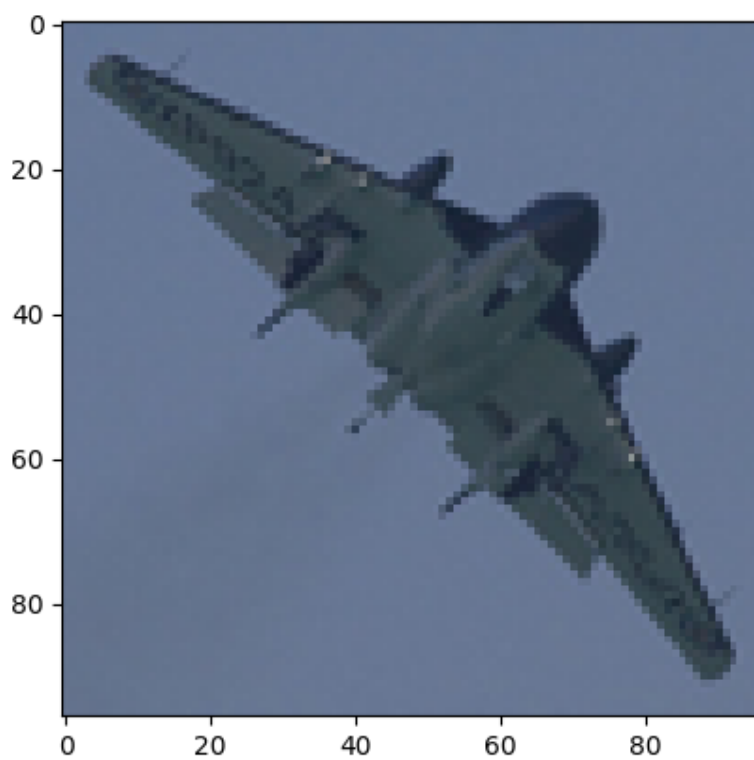


Un exemple de résultat après 13 époques :

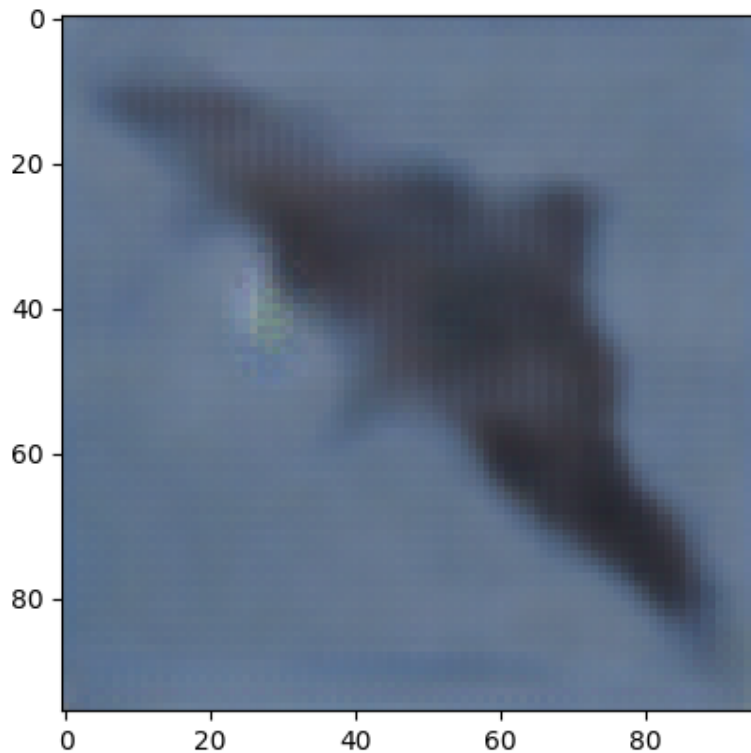
- Image de base



- Image attendue

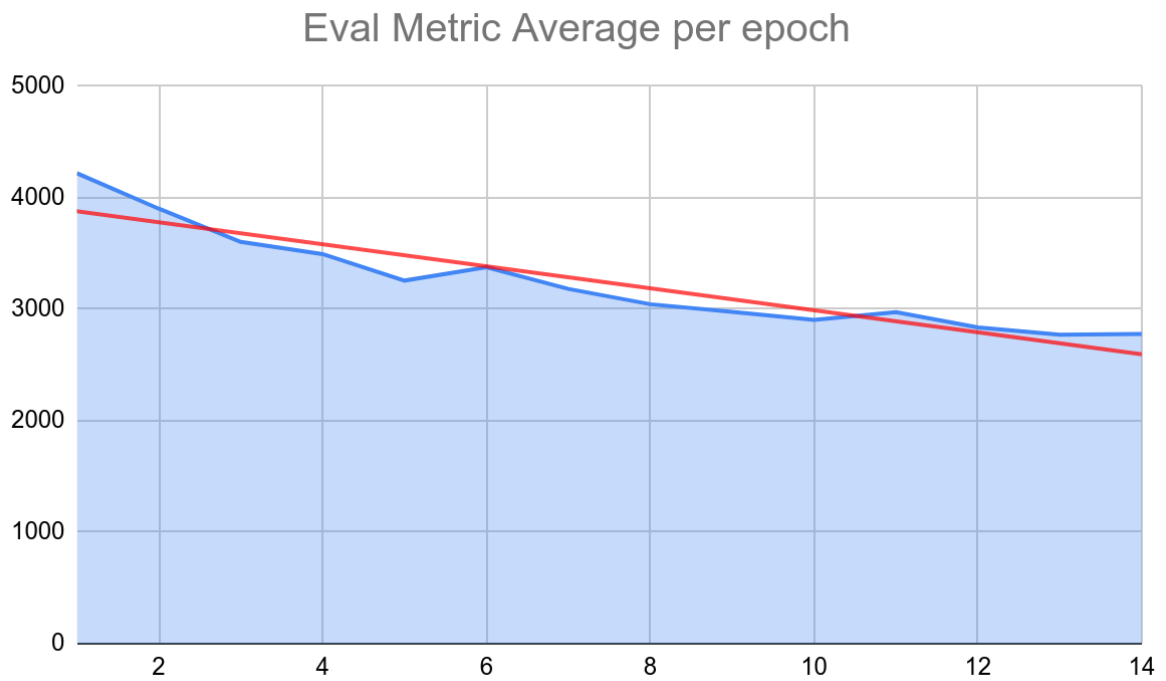


- Image prédite

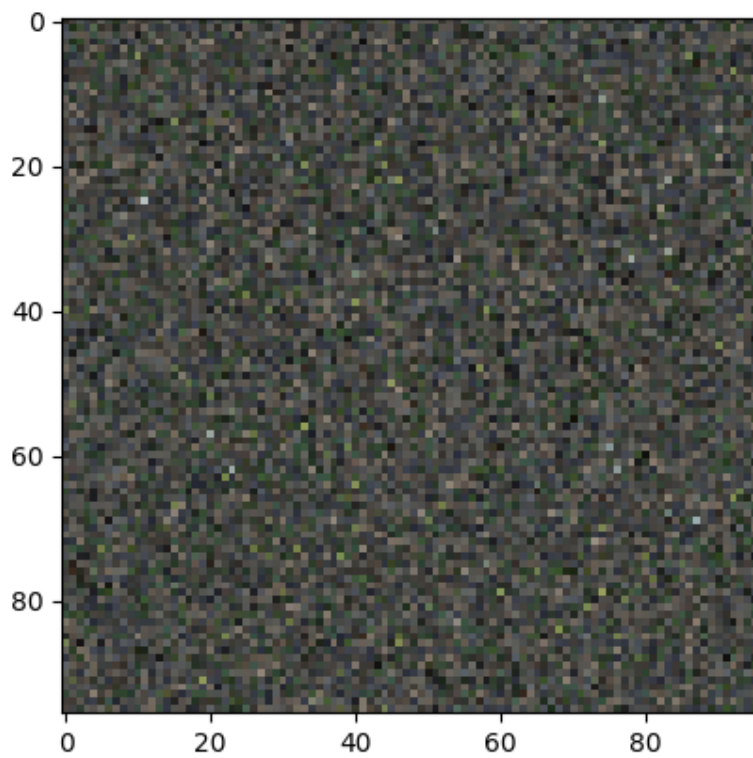


Partie 2 – Resnet18 :

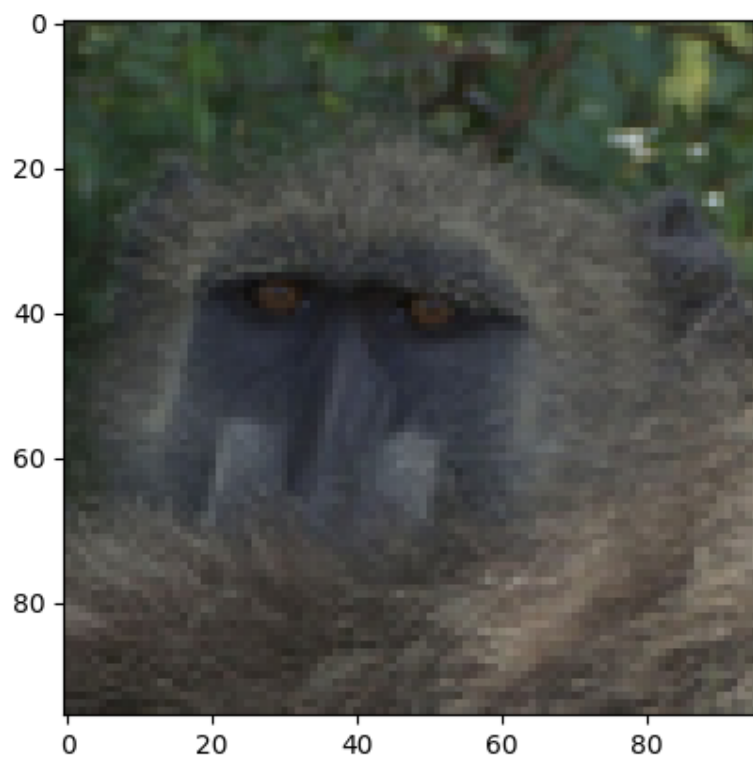
Pour cette partie nous avons réussi à obtenir un delta moyen de 2771 au minimum lors de l'epoch numéro 13. Sur l'image prédite, on peut observer la silhouette d'un singe sans en distinguer les détails. Lorsque l'on regarde la courbe d'évolution de l'évaluation, on peut affirmer qu'il est encore possible d'améliorer modérément les résultats. Si la tendance reste la même, on peut considérer que nous arriverons à avoir une image légèrement plus détaillée.



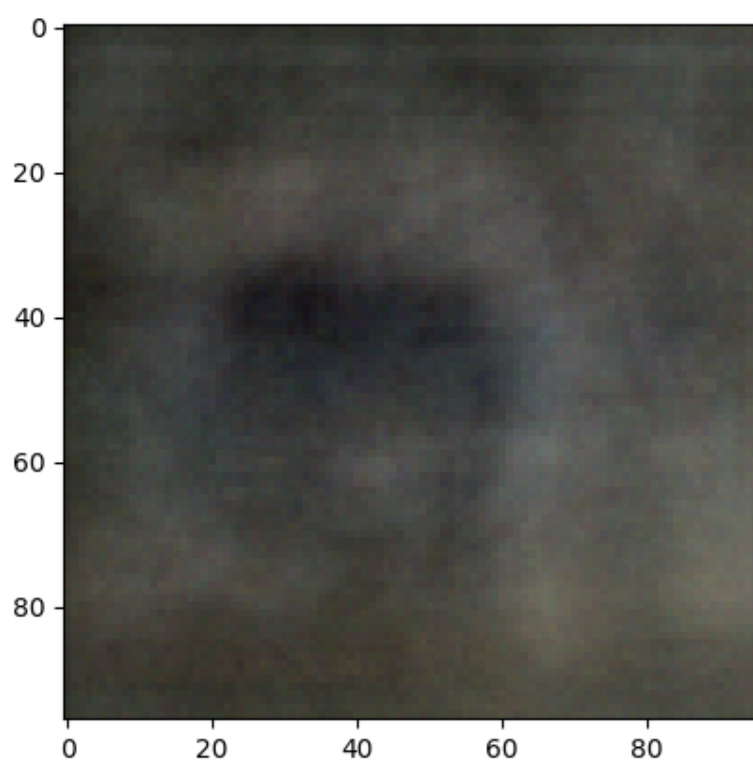
- Image de base :



- Image attendue :



- Image prédite :



Conclusion

Pour conclure, nous voyons que le modèle ResNet 18 ne permet pas d'obtenir rapidement des résultats significatifs. Cependant, le modèle U-Net que nous avons établi permet d'obtenir de bons résultats sur la partie restauration d'image. On se retrouve avec une image que l'on peut totalement reconnaître même si on distingue toujours des imperfections dans certains détails.

Néanmoins, concernant le déchiffrement les résultats sont plus mitigés, on observe un début de forme reconnaissable, mais pas assez pour être correcte comme pour la restauration.

Nous avons aussi été limités au niveau des ressources matérielles. Nous avons fait le choix d'utiliser un conteneur docker pour lancer nos programmes et aucun d'entre nous n'avait de GPU, ce qui a limité le nombre d'epochs que nous pouvions lancer pour les tests. Peut-être que sur plus d'epochs nous aurions pu obtenir d'encore meilleurs résultats, notamment sur la partie déchiffrement.