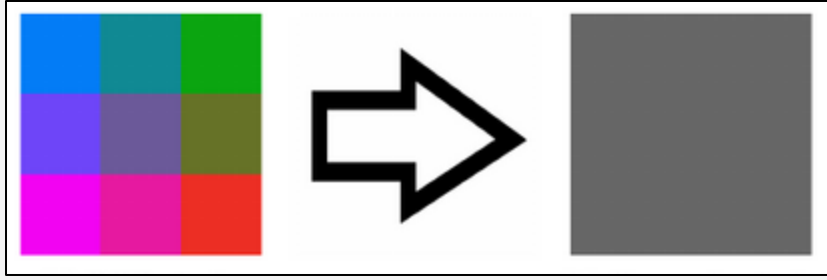


Colorisation d'images

Pourquoi ?



- Il existe des règles pour convertir une image RVB en noir et blanc
- Par contre, beaucoup de couleurs correspondent à un même niveau de gris
- L'apprentissage profond nous permet d'estimer ces couleurs à l'aide de patterns détectés dans l'image.

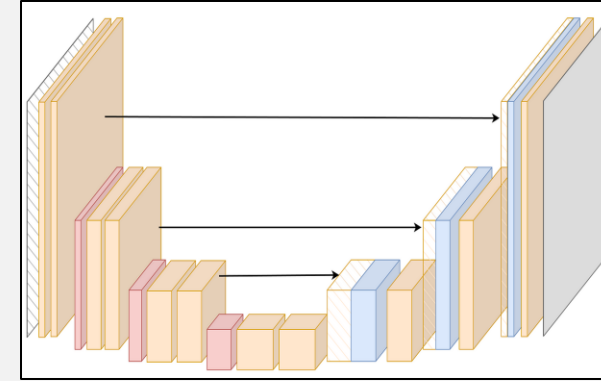
Limites du modèle

- Le modèle a été entraîné sur 20 000 images de paysage au format 128x128
- Nous avons été contraints par la quantité de mémoire et la puissance de calculs de nos ordinateurs
- La faible résolution rend difficile la détection de certains petits patterns.
- Le modèle est uniquement entraîné à coloriser des paysages

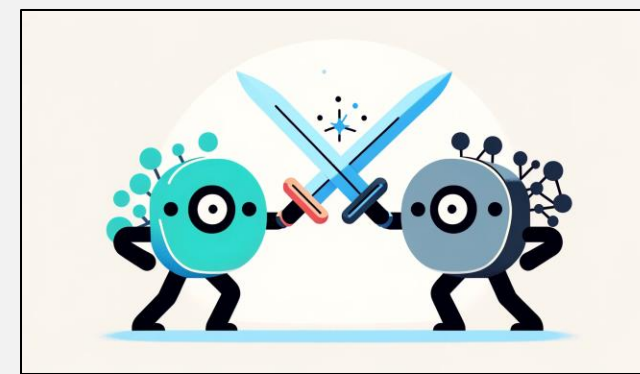


- L'utilisation d'un échantillon beaucoup plus conséquent et d'images plus larges permettrait une performance nettement supérieure
- Le portage du modèle de TensorFlow à PyTorch pourrait permettre l'utilisation d'espaces de couleurs plus adaptés à la colorisation

Comment ?



- Notre premier modèle est un U-net, une variante des autoencodeurs
- Un modèle pré-entraîné est utilisé comme encodeur pour faciliter l'entraînement
- Le modèle tente de recréer l'image en couleur à l'aide de la même image en noir et blanc



- L'utilisation d'un réseau antagoniste génératif (GAN) permet d'affiner le résultat du U-net
- Le U-net est entraîné à tromper un discriminateur. Le discriminateur est entraîné à reconnaître les images générées
- Ce nouvel objectif permet au U-net d'explorer de nouveaux espaces, ce qui améliore la colorisation

Performance



- L'utilisation de métriques n'est pas très pertinente dans le cas de la colorisation
- Une inspection visuelle nous permet de voir une amélioration de la fidélité entre le U-net initial et celui issu du GAN dans la plupart des cas
- Puisque les différences sont trop subtiles pour être visibles une fois imprimées, les résultats sont observables sur l'ordinateur derrière vous