

Marius Jenin
Alexandre Lanvin



Projet IMAGE

CR6

Avancement

Jusqu'ici nous avons implémenté une application de cartoonisation d'image sans réseaux de neurones et deux transferts de styles différents (l'un utilise *keras* l'autre utilise *pytorch*). Nous nous sommes ainsi basé sur ce [papier](#) [1] dont les résultats sont prometteurs lorsque l'on applique un style assez abstrait comme de la peinture :



Figure 1: Transfert de style issu du papier

Voici le résultat obtenu avec notre implémentation sur Keras en utilisant les mêmes images de contenu et de style :



Figure 2: Transfert de style sur Keras (100 itérations)

Nous avons alors tenté d'appliquer cette méthode en utilisant une image de style cartoon, voici le résultat obtenu :

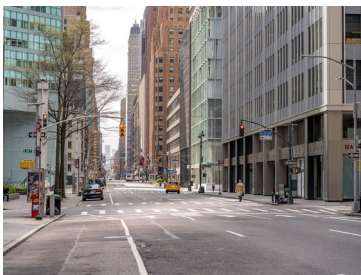


Figure 3: Image de référence

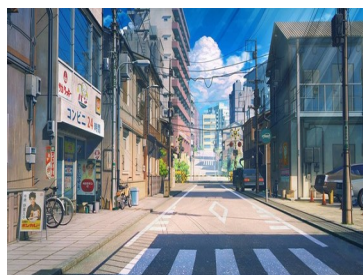


Figure 4: Image de style

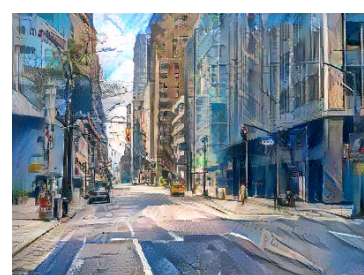


Figure 5: Image produite (4000 itérations)

[1] Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576.

[2] Chen, Y., Lai, Y. K., & Liu, Y. J. (2018). Cartoongan: Generative adversarial networks for photo cartoonization. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 9465-9474).

Nous avons alors constaté que les images produites avec une image de style *cartoon* comportent trop d'artefacts même en faisant un grand nombre d'itérations. Parfois on ne retrouve pas vraiment une image avec de bons contours et aplats de couleurs qui sont pourtant caractéristiques du style cartoon.

Bien que le transfert de style soit une méthode efficace et ne nécessitant pas d'entraînement d'un modèle (*nous utilisons vgg19 avec les poids de imagenet*) ou d'un grand nombre de données (*seulement une image de contenu et une image de style*) il nous est apparu nécessaire d'explorer d'autres méthodes pour obtenir une meilleure cartoonisation d'images utilisant les réseaux de neurones, plus particulièrement les GANs.

La méthode que nous avons choisi pour utiliser un GAN est décrite dans ce [papier](#) [2] qui propose alors de construire un *cartoonGAN*.

Voici quelques résultats de cartoonisation d'images présentés dans le papier :



Figure 6: Image de base à gauche, image stylisée à droite

[1] Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576.

[2] Chen, Y., Lai, Y. K., & Liu, Y. J. (2018). Cartoongan: Generative adversarial networks for photo cartoonization. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 9465-9474).

Après avoir récupéré des poids d'un *cartoonGAN* déjà entraîné nous avons pu obtenir quelques résultats :



Figure 7: Résultats avec le *cartoonGAN* déjà entraîné

Pistes d'améliorations

- Il est possible d'utiliser plusieurs images de style dans le transfert de style ce qui pourrait permettre d'obtenir une meilleure image cartoonisée en sortie.
- Entraîner à nouveau le *cartoonGAN* récupéré pour pouvoir appliquer un style différent de celui qui est à présent utilisé.

[1] Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576.

[2] Chen, Y., Lai, Y. K., & Liu, Y. J. (2018). Cartoongan: Generative adversarial networks for photo cartoonization. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 9465-9474).