SUPER RÉSOLUTION

stage d'été 2018-2019

Auteurs: BEROUKHIM Keyvan
RUEL Paul

Encadrant: GALLINARI Patrick

I - Introduction

La super-résolution est le processus qui consiste à créer une image en haute qualité à partir d'une ou plusieurs images en basse qualité. Durant ce stage, nous nous focalisons sur la tâche **SISR** (« Single Image Super Resolution ») en se servant de **GAN** (« Generative Adversarial Network »).

à partir d'une image en basse résolution, notre réseau de neurone génère une image de plus haute résolution

II - SRGAN

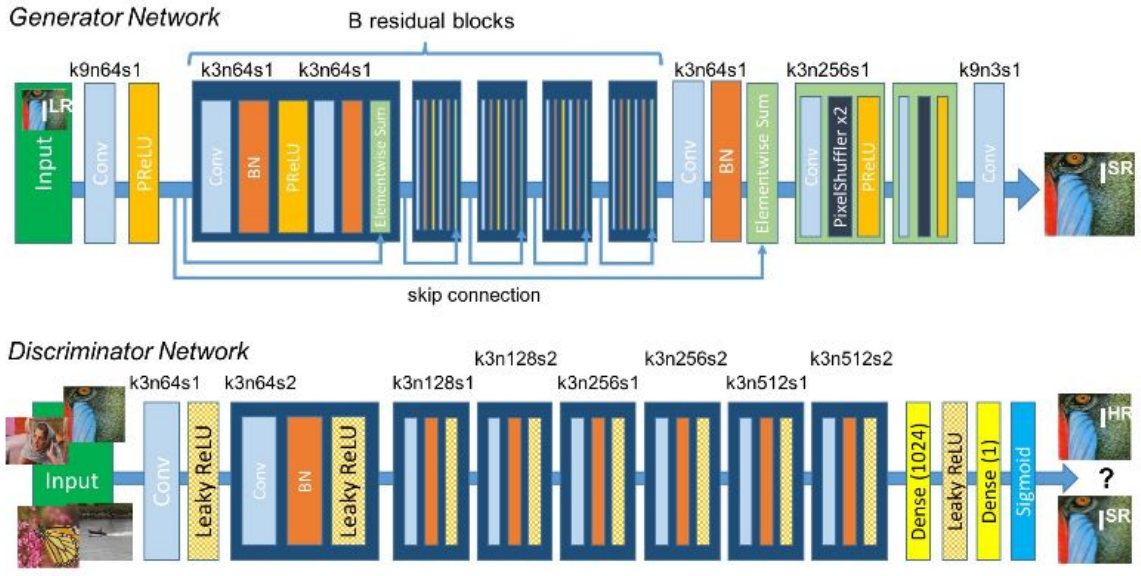
Pour travailler en super-résolution, nous nous sommes appuyés sur le modèle SRGAN présenté dans l'article [1], état de l'art en super-résolution à sa sortie, et avons essayé d'en reproduire les résultats. En se servant d'un GAN avec des blocs résiduels, SRGAN génère des images avec une échelle x4 en taille (c'est-à-dire avec 16 fois plus de pixels).

L'apport de l'article réside dans la combinaison du **réseau génératif adverse** et d'une **fonction de coût "perceptuelle"** (perceptual loss ou content loss). Dans sa forme la plus simple, le coût perceptuel est une MSE entre les valeurs des pixels de l'image générée et de l'image originelle.

En utilisant seulement un coût perceptuel, le générateur a tendance à conduire à une moyenne des solutions les plus plausibles et donc à une sortie floue. L'ajout du coût adversaire permet de pallier à ce problème en rendant les images générées indistinguables des images réelles, c'est à dire nettes.

Réciproquement, en utilisant seulement un coût adversaire, il n'y a pas de contraintes forçant les images générées à ressembler aux images en entrées. Ajouter un coût perceptuel permet alors de pallier à ce problème.

Plutôt que d'utiliser une MSE entre les pixels des images, une proposition est d'utiliser une représentation plus haut niveau de cette image. Dans cet article, la représentation de haut niveau est obtenue en utilisant une couche latente du réseau VGG.



Architecture des réseaux générateur et discriminateur (schéma tiré de l'article SRGAN)

III - Notre implémentation

1) Extensions/Améliorations du modèle SRGAN

- **Initialisation de G** en utilisant seulement une **MSE pendant une époque**. Cela permet à G de **converger rapidement** vers des images générées proches du résultat final attendu.
  Poursuivre par l'entraînement synchronisé habituel de G et D détériorerait l'état de G car il se base sur la sortie de D qui est initialisé aléatoirement. On entraîne donc **D uniquement** pendant quelques itérations (moins d'une époque pour ne pas sur-apprendre l'état actuel de G).

- **Augmentation graduelle du poids de la loss adversaire** (par rapport à la content loss) tant que cela améliore le rendu. Cette méthode **assure des résultats corrects**, c'est à dire au moins aussi bon que ceux obtenus par les réseaux se contentant de minimiser la MSE.

- **'One-sided label smoothing**': Les vraies images sont présentées au discriminateur avec un label (ou 'score de réalisme') valant **0.9 au lieu de 1** afin que D ne soit pas sûr de lui, cela permet de stabiliser l'apprentissage.

- **'Experience replay**': Sauvegarde des **anciennes images générées** pour les **re-présenter à D lors des époques suivantes**. Cela permet à chaque itération que D ne ne pas sur-apprenne pas l'état temporaire de G.

- Utilisation de **'SpectralNorm**' pour le générateur et le discriminateur. Cette couche permet de normaliser les matrices de poids du réseau afin de le rendre plus résistant aux perturbations.

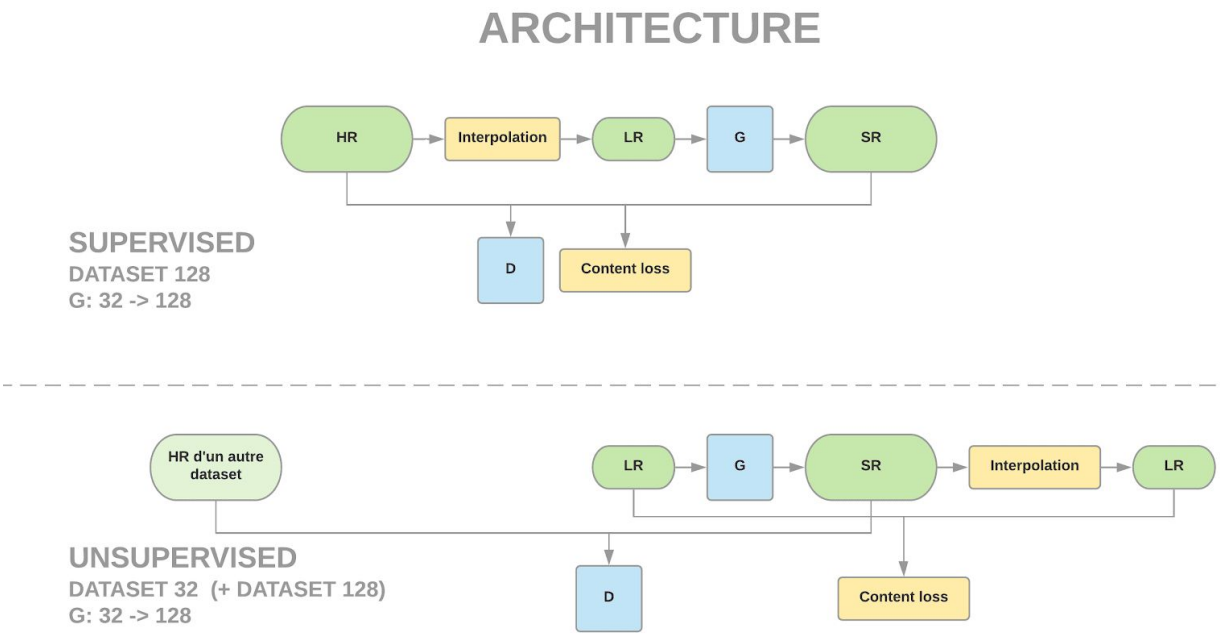
- Un facteur d'agrandissement x4 est obtenu en appliquant deux fois un agrandissement x2. La structure des deux réseaux est fortement similaire, **90% des poids d'un réseau x4 peuvent être chargés à partir d'un réseau x2**. **'Progressive GAN'**: on commence par entraîner un réseau x2, on ajoute ensuite une couche d'agrandissement à la fin du réseau x2 afin de créer un réseau x4. Un réseau créé de la sorte est bien meilleur qu'un réseau x4 entraîné 'from scratch' pour une même durée d'entraînement. Par ailleurs, on peut geler toutes les couches à part celle rajoutée afin d'aller plus vite.

- Extraction des 'features map' de VGG avant passage par la couche d'activation, cela permet de ne pas perdre de signal à cause de l'activation.

- Le calcul de LR par interpolation bicubique de HR dépasse un peu de l'intervalle [-1, 1]. On seuille les images pour les remettre dans [-1, 1].

- En partant d'un même réseau et avec un même dataset non mélangé, deux entraînements donnent des images complètement différentes, on peut résoudre ce problème en changeant les valeurs de torch.backends.cudnn.deterministic et torch.backends.cudnn.benchmark.

2) Modèle non supervisé



L'architecture de SRGAN a besoin que les images sur lesquelles on entraîne le réseau soient en HR. Nous proposons un modèle n'ayant pas cette contrainte. Pour la content loss le modèle se sert de la version LR des images. Pour la loss adversaire, le discriminateur peut utiliser des images HR d'un autre dataset. Plusieurs problèmes se posent :

- Pour le calcul de lu coût perceptuel, sous-échantillonner l'image SR fait perdre beaucoup d'informations, **cela rend le réseau plus difficile à entraîner**. La technique consistant à augmenter graduellement le poids du coût adversaire n'est plus très opportune car avec le coût perceptuel uniquement, les images générées sont de très mauvaise qualité.
- Pour la loss adversaire, si les datasets ne sont pas semblables, le discriminateur risque de différencier les images par le contenu (le type d'objet représenté) plutôt que par la qualité.

IV - Résultats

Une fois le réseau entraîné, nous lui présentons des images afin de visualiser ses performances. Les résultats sont présentés sous la forme suivante:
1ère ligne : LR [32] SR=G(LR) [128] HR [128] UR=G(HR) [512]
2ème ligne : ce sont les mêmes images que celles de la premières mais étirées d'une autre manière (interpolation bicubique).

Les images en Ultra-Résolution sont les images obtenues en appliquant le réseau aux images en haute qualité, elles montrent la capacité du réseau à s'appliquer à des images de plus haute résolution.

1) supervisé x4

Le réseau a été entraîné à générer des images de taille 128x128 à partir d'images de taille 32x32.

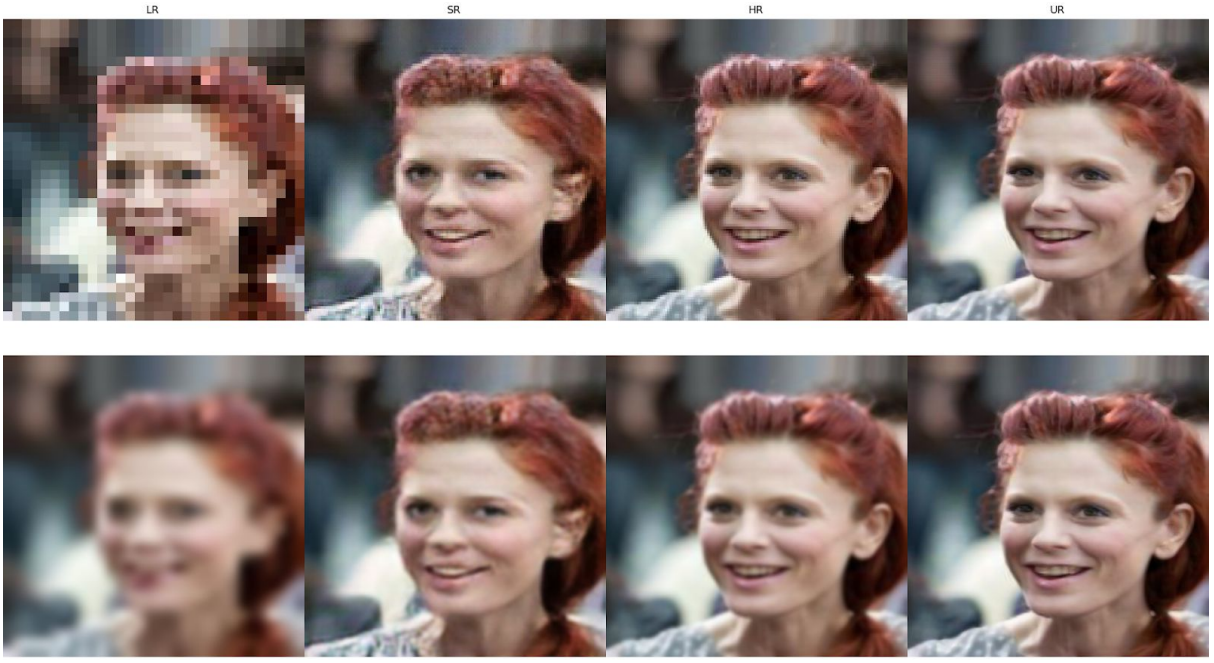


Image issue du dataset d'apprentissage: le réseau de neurones génère une image de bien meilleure qualité que l'image d'entrée

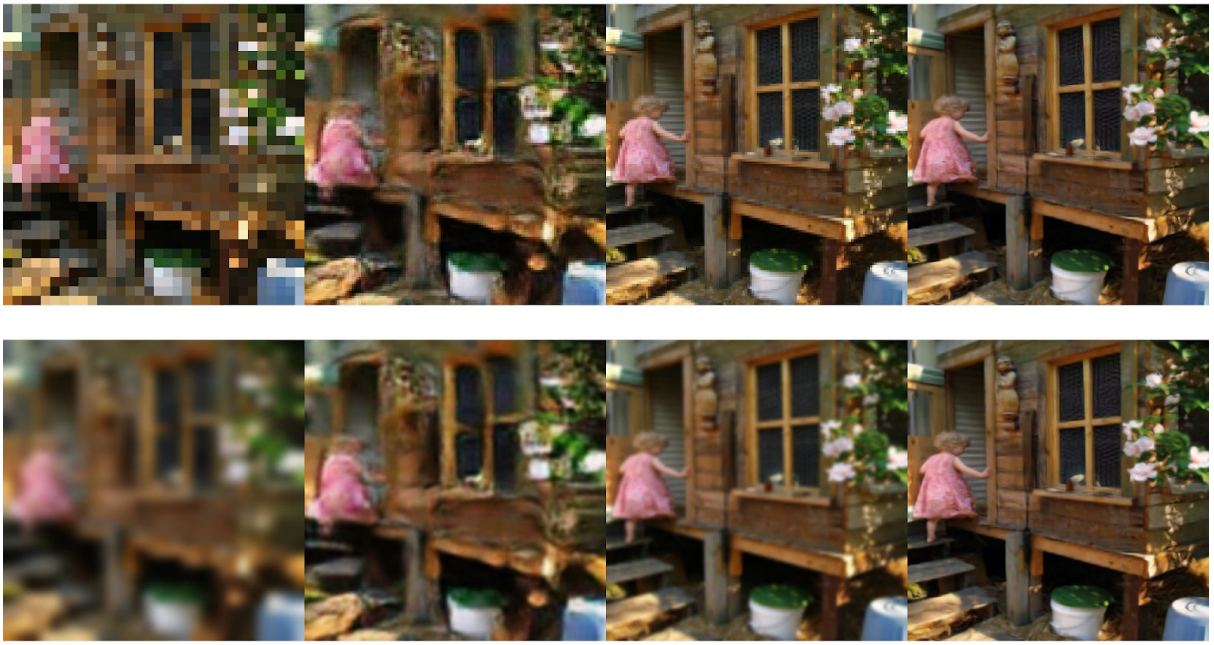


image issue d’un autre dataset: à la différence des images d’entraînement, cette image ne représente pas un visage. Le réseau de neurone parvient néanmoins à générer une image de bien meilleure qualité

image dégradée différemment, le réseau de neurone génère une image super-résolue de mauvaise qualité.

2) supervisé x8 sur CelebA

Les images de CelebA (128x128) sont trop petites pour un agrandissement x8 car des images en basse résolution 16x16 ne contiennent pas assez d'information.

Avec une MSE seulement, le résultat est flou. Avec un coût adversaire le GAN invente un visage (ou plusieurs, comme dans la capuche rouge de l'image en haut à droite)

3) non supervisé x2

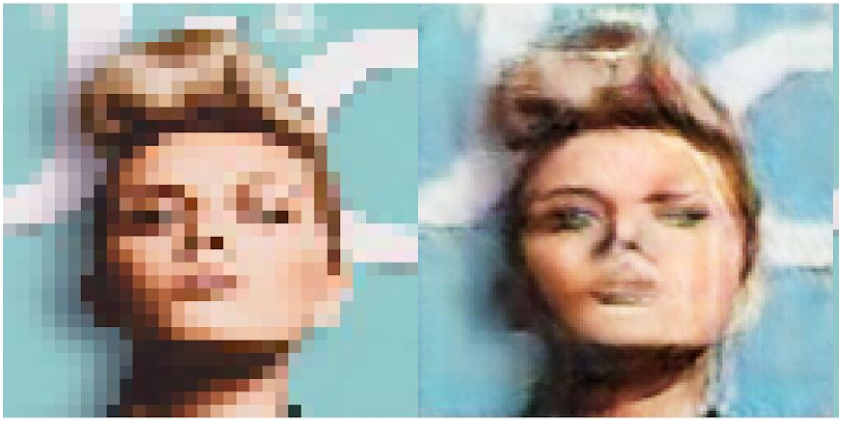
Le réseau de neurones génère une image deux fois plus résolue

En entraînant le réseau à générer des images 128x128 à partir d'images 64x64, le réseau parvient à générer des images de meilleurs qualité. On pourrait sûrement améliorer les résultats en entraînant le réseau plus longtemps, mais on a préféré se consacrer à de plus grands agrandissements.

4) non supervisé x4

a) Premières tentatives

Les premières tentatives d'entraînement non supervisé x4 n'ont pas abouti. Voici des exemples d'images générées :



En entraînant le réseau 32->128 sur le dataset Celeba, on peut supposer que les images LR ne possèdent pas suffisamment d'informations pour le coût perceptuel.

Nous avons tenté de prendre des images deux fois plus résolues (donc 64->256) afin qu'il y ait plus d'informations dans les images LR (nous nous sommes servis du dataset Flickr dont les images sont de base légèrement plus grandes que 256x256), mais les résultats n'ont pas été meilleurs.

Avec un poids du coût adversaire nul, les images générées sont quasiment identiques aux entrées. En augmentant graduellement le poids, des artefacts apparaissent dans les images générées.

b) Résultats corrects

Au cours de nos expérimentations, sans ajouter de nouvelles améliorations, mais en faisant varier les valeurs des hyperparamètres, nous sommes tombés sur des reconstructions correctes.

exemple de reconstruction correcte du réseau non supervisé x4

Un problème rencontré est qu'en continuant l'entraînement, les images générées **oscillent mais ne s'améliorent pas**. Afin de continuer l'entraînement, nous avons tenté de baisser le learning rate et le poids de la loss adversaire et d'augmenter le nombre de batchs sauvegardés puis re-présentés à D. Le problème qui apparaît alors est, d'une part, que le fait de baisser le learning rate et d'augmenter le nombre de batchs sauvegardés ralenti l'entraînement et d'autre part, que baisser le poids de la loss adversaire rend les images générées floues. Les images générées variant d'une exécution à l'autre et progressant très lentement, il est difficile d'améliorer les résultats obtenus, nous nous approchons des **limites du modèle**.

V - Conclusion

Durant ce stage nous avons implémenté des techniques de super-résolution se servant de GAN. Ce stage a été très formateur, nous avons acquis de nombreuses connaissances sur les réseaux de neurones, les GAN et la super-résolution en lisant des articles de recherche; et par la même occasion, nous nous sommes habitués à lire des articles de recherche. De plus, nous avons amélioré notre maîtrise de PyTorch et avons acquis une bonne méthodologie pour lancer des expériences.

Travailler avec des images 1024x1024 est au moins 64 plus long que de travailler avec des images 128x128. Certains problèmes auraient peut-être pu être résolus avec de grandes images mais nous ne pouvons pas travailler avec de trop grandes images pour des raisons de temps de calcul.

Entraîner un GAN est un problème difficile, mais c'est heureusement un problème auquel beaucoup de gens font face et de nombreux travaux de recherche proposent des techniques aidant à l'entraînement du réseau. On peut supposer que l'on arrivera bientôt à générer de meilleures images quatre fois plus grandes (ou même plus) dans un contexte non supervisé.

VI - Remerciements

Nous remercions toute l'équipe MLIA de nous avoir accompagné (et prêté leur matériel) : notre encadrant ainsi que les autres professeurs, les doctorants, et même les autres stagiaires, qui nous ont apporté conseils et motivation.

VII - Bibliographie

Notre GitHub (contenant notre carnet de bord)
https://github.com/keyber/Single-Image-Super-Resolution

RÉSEAUX DE NEURONES

1) SRGAN https://arxiv.org/abs/1609.04802
2) GAN https://arxiv.org/abs/1406.2661
3) VGG https://arxiv.org/abs/1409.1556
4) DCGAN https://arxiv.org/abs/1511.06434
5) ESRGAN https://arxiv.org/abs/1809.00219
6) ProgressiveGAN https://arxiv.org/abs/1710.10196
7) BatchNorm https://arxiv.org/abs/1502.03167
8) SpectralNorm https://arxiv.org/abs/1705.10941

BASES DE DONNÉES

1) MNIST http://yann.lecun.com/exdb/mnist/
2) Celeba http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
3) flickr https://forms.illinois.edu/sec/1713398
4) flickr_HQ_faces https://github.com/NVlabs/ffhq-dataset