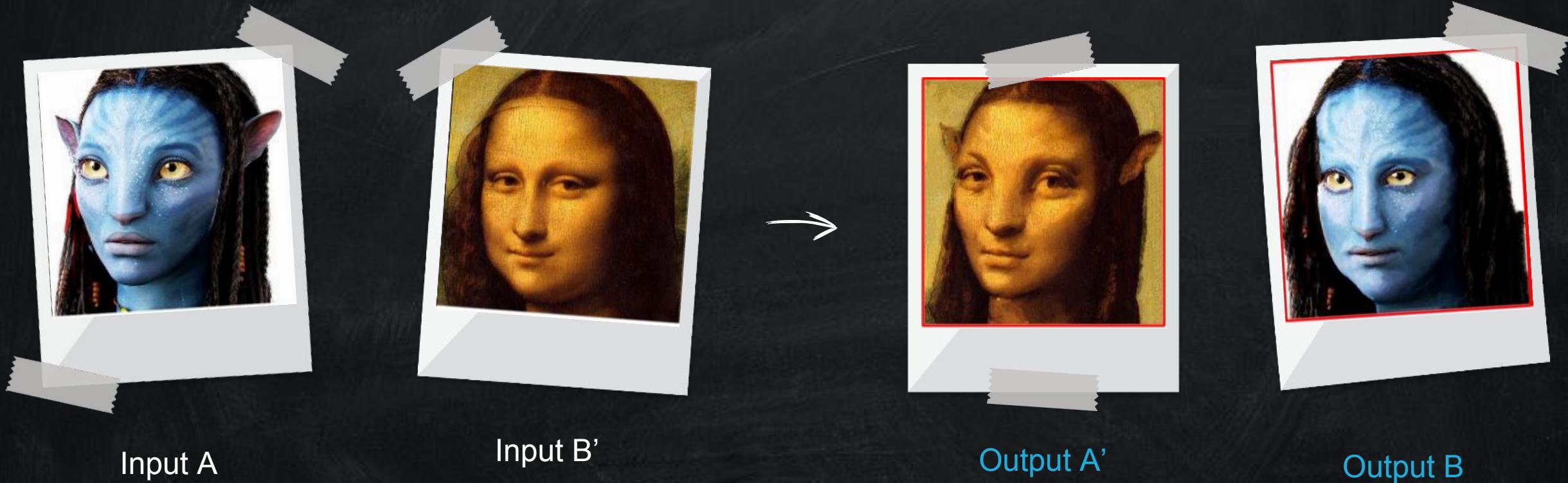


Visual Attribute Transfer through Deep Image Analogy

Jing Liao, Yuan Yao², Lu Yuan, Gang Hua, and Sing Bing Kang
Microsoft Research, Shanghai Jiao Tong University



Input A

Input B'

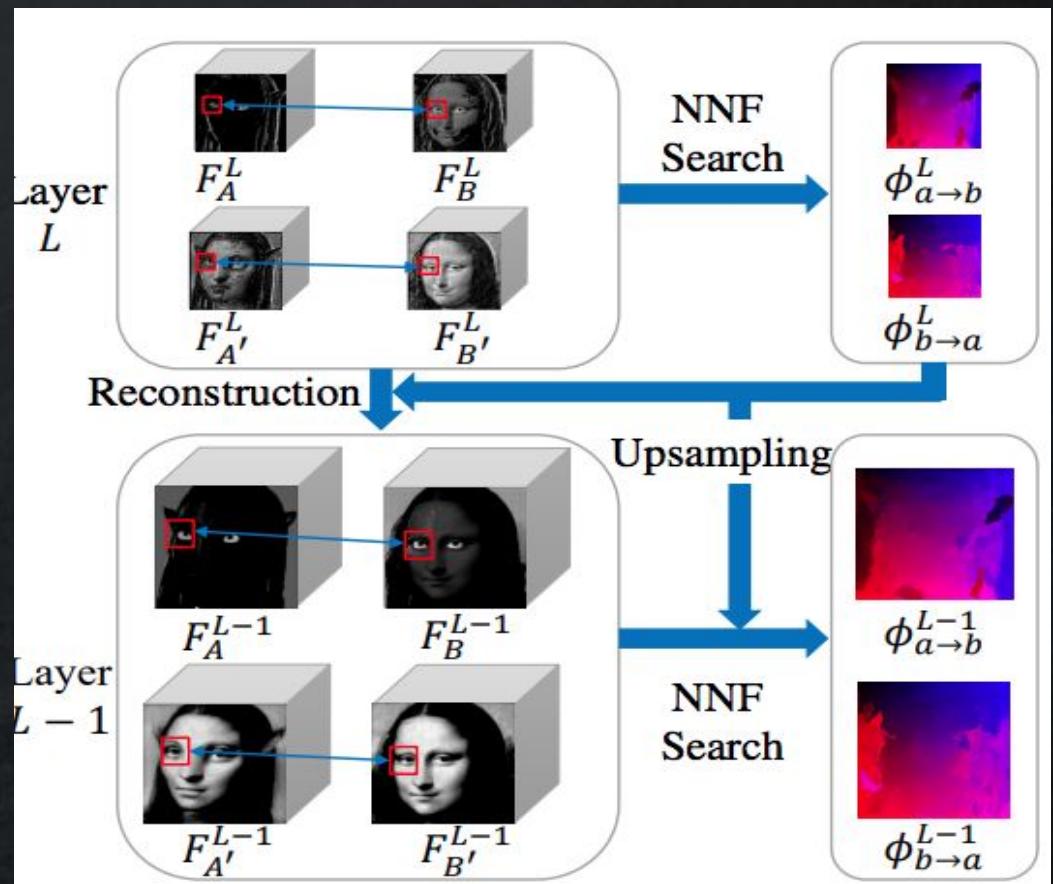
Output A'

Output B



Rappel

- Algorithme Itératif sur 5 couches du vgg
- On possède au début les FA et les FB'
- A chaque itération :
 - Une phase de Mapping
 - Une phase de reconstruction
(pour obtenir FA' et FB à la couche considérée)



Problème



Input A

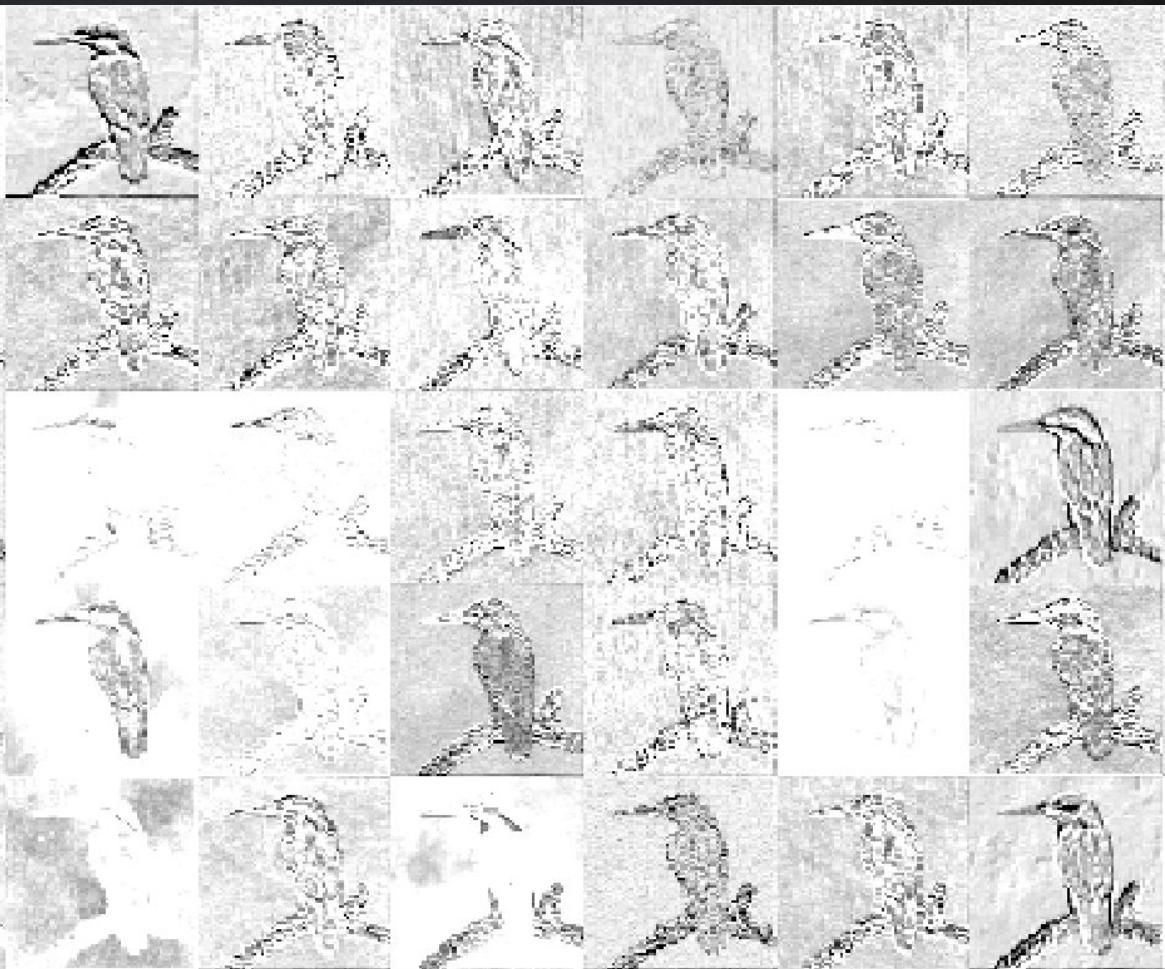


Input B'

Création des Features



Image d'entrée



Sortie à la couche relu2_1 du vgg19

Deep PatchMatch

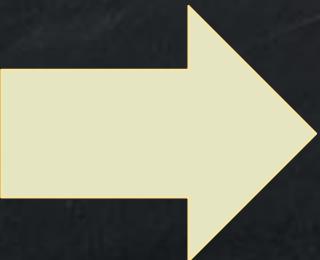
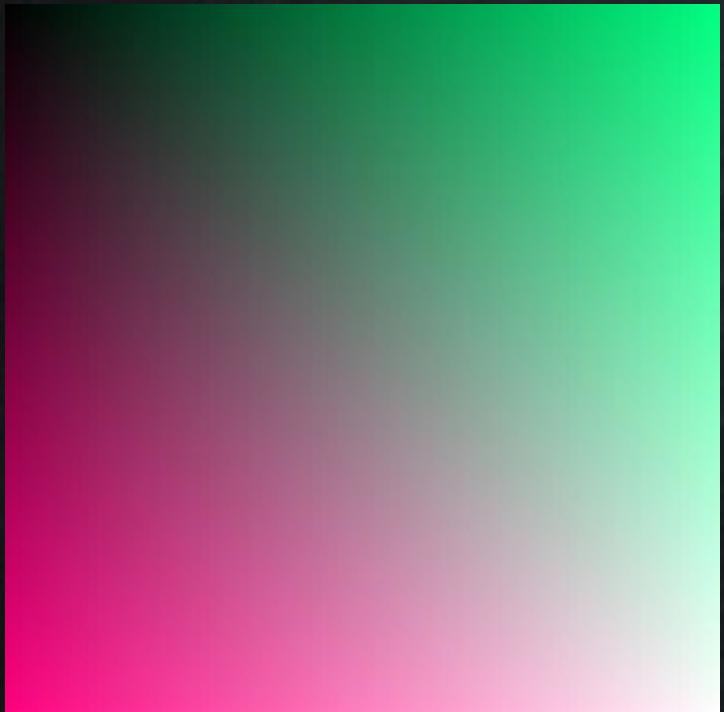
Objectif : Mapping Sémantique sur les DeepFeatures

Différences avec un PatchMatch Classique :

- Accepte des matrices 3D de profondeur arbitraire
- Utilise une initialisation pour le mapping
- Distance utilisant 4 matrices (cf Image Analogy)
- Randomisation moins importante

$$\phi_{a \rightarrow b}^L(p) = \arg \min_q \sum_{x \in N(p), y \in N(q)} (\|\bar{F}_A^L(x) - \bar{F}_B^L(y)\|^2 + \|\bar{F}_{A'}^L(x) - \bar{F}_{B'}^L(y)\|^2),$$

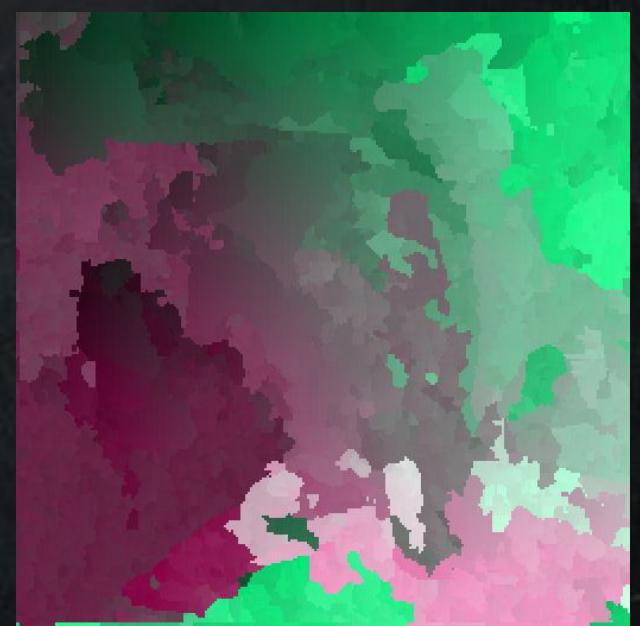
Deep PatchMatch



Mapping à la couche 4



Mapping à la couche 2



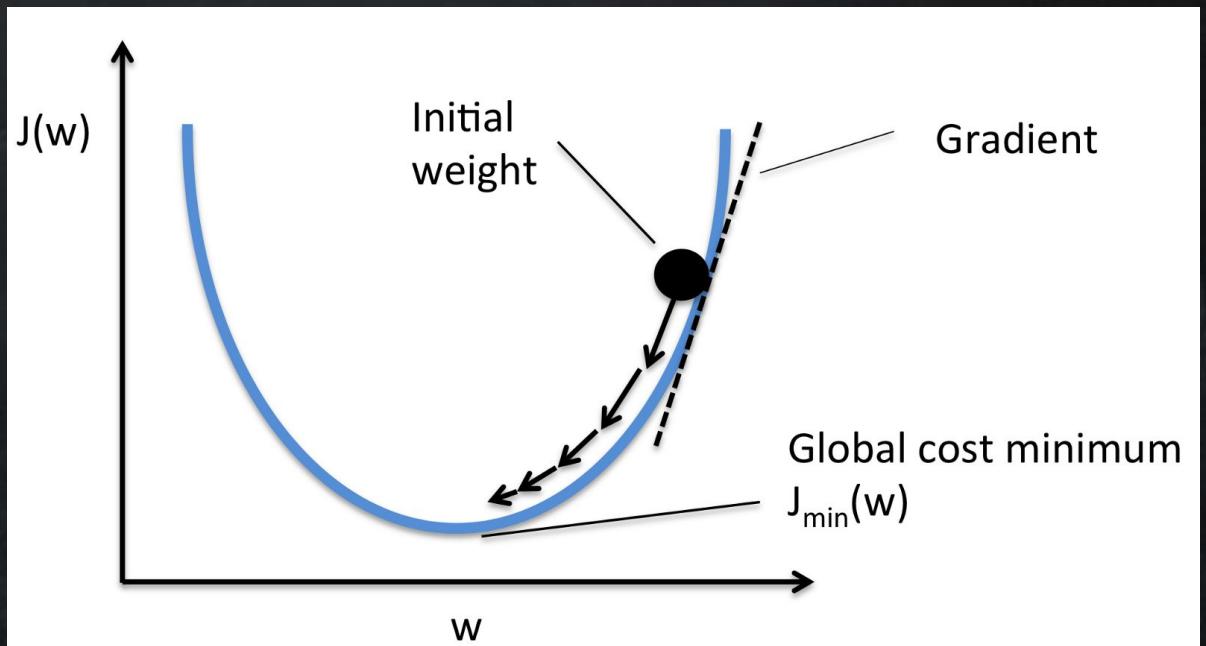
Descente de gradient

Objectif : Optimiser le matrice mapping

utilisée pour la reconstruction

Basé sur l'algorithme: L-BFGS-B

- + Simplifier le calcul de step
- + Paralléliser le calcul sur une matrice 3D
- + Accepter les tailles de matrice différentes en entrée



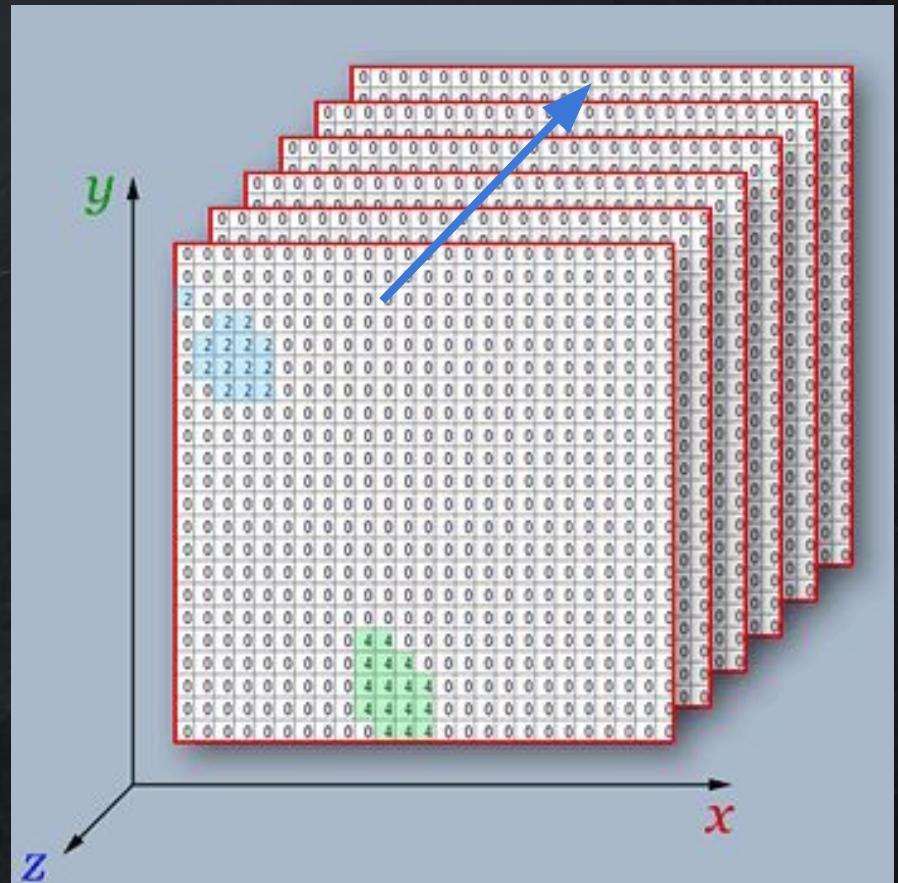
Descente de gradient

$$R_{B'}^{L-1} = F_{B'}^{L-1}(\phi_{a \rightarrow b}^{L-1})$$

$$\mathcal{L}_{R_{B'}^{L-1}} = \|\text{CNN}_{L-1}^L(R_{B'}^{L-1}) - F_{B'}^L(\phi_{a \rightarrow b}^L)\|^2$$

Problème:

- + L'optimisation est fait sur chaque vecteur profondeur indépendamment, mais le calcul de gradient n'est possible que sur l'ensemble de matrice



Descente de gradient

iteration 2 couche 1



iteration 2 couche 10



iteration 4 couche 1



iteration 4 couche 10



Reconstruction



FA'(1) à la couche 2



FB(1) à la couche 2

Reconstruction



Résultat final



Résultat final

Evolution en fonction du paramètre alpha



Augmentation d'alpha



Résultat final

Evolution en fonction de la randomisation du DeepPatchMatch

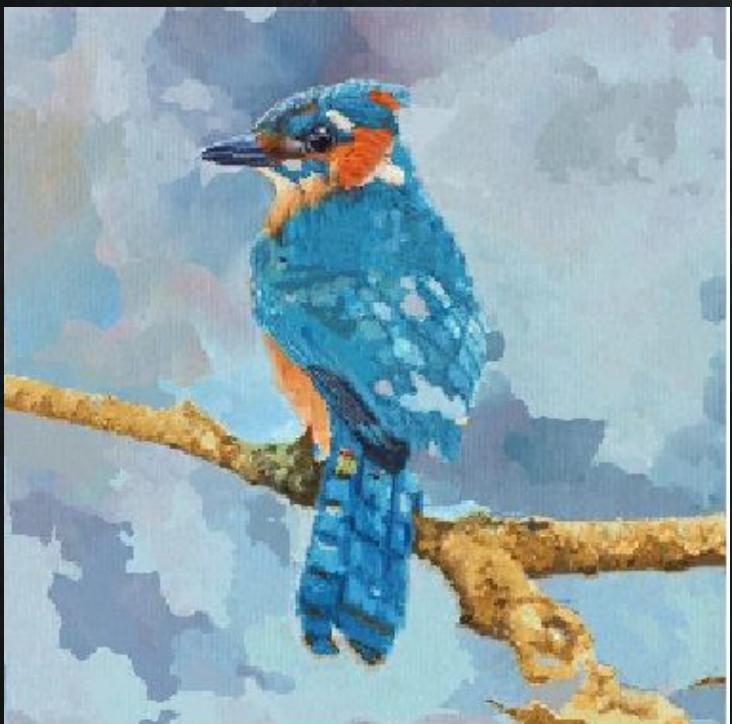


Augmentation de la randomisation



Résultat final

Nombre d'itérations (= Nombre de couches du vgg19 utilisées)



$L = 5$



$L = 4$



$L = 3$

Comparaison avec l'article

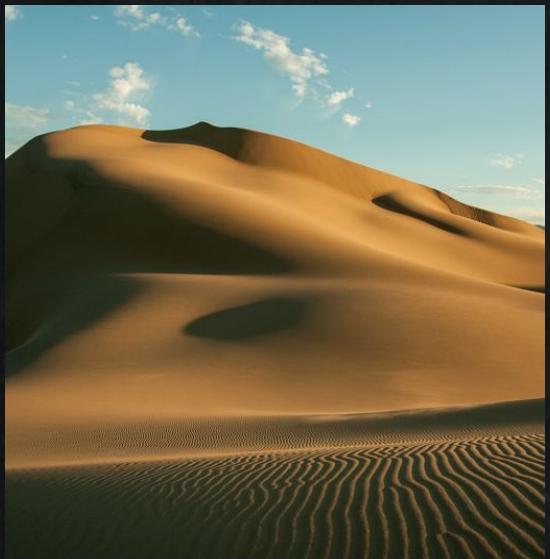


Critiques

- Relativement lent surtout lorsqu'on augmente la taille des images
- Problèmes avec les régions sémantiquement pauvres
- Erreur si une unité sémantique d'une image ne se trouve pas dans l'autre
- Nécessite de nombreux ajustements de paramètres
- Le réseau de neurones ne conserve pas la localité des unités sémantiques (Cf. Max pooling, ReLU)
- L'opération de l'optimisation est coûteuse (temps d'exécution important)
- + Nombreuses applications pour le mapping sémantique
- + Idée réutilisable pour d'autres réseaux et d'autres applications



Autres résultats



Autres résultats



Autres résultats



Questions?

