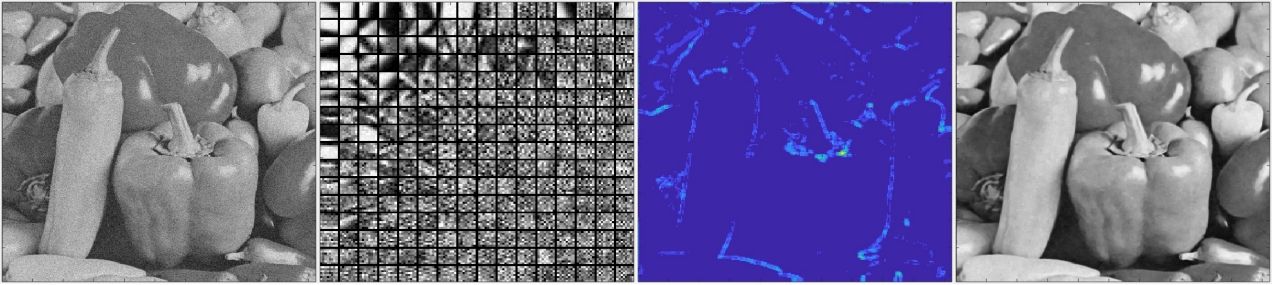


TP d'Introduction à l'Apprentissage Automatique

Exo 5: Apprentissage de Dictionnaire avec K-SVD

Télécom Physique Strasbourg - antoine.deleforge@inria.fr



Le but de cet exercice est de comprendre comment la reconstruction parcimonieuse peut être appliquée au débruitage d'image, et comment apprendre un dictionnaire à partir de morceaux d'image peut améliorer les résultats. Pour cela, vous implémenterez l'algorithme K-SVD [1], et comparerez le dictionnaire appris au dictionnaire de la Transformée en Cosinus Discrète (*discrete cosine transform* ou DCT). Les fichiers du TP (Matlab) sont à télécharger en suivant ce lien : members.loria.fr/ADeleforge/files/TP_ML_Exo5_TPS.zip.

Partie I : Débruitage d'Image avec la DCT

1) Lancez le script principal `TP_DICTLEARN_denoise_script.m` et observez les figures produites. Que fait ce script? Quelle est la dimension de la variable `dict` et que contient-elle? Que signifie le terme PSNR? Cherchez dans le code comment cette valeur est calculée.

2) Derrière l'image du dictionnaire se cache une figure représentant le nombre d'atomes de dictionnaire utilisés par pixel. Quelles zones de l'images requièrent le plus/le moins d'atomes, et pourquoi? Changer d'image à débruiter et observez les résultats.

Note : les versions `_small` des images sont uniquement fournies pour le cas où vous rencontreriez des problèmes de mémoire ou de temps de calcul lors du TP.

3) A quoi correspond le paramètre `dictsize`? Essayez de le changer à 64 et à 1024 pour voir l'impact sur le dictionnaire. Y'a t'il un impact significatif sur le nombre d'atomes utilisés par pixel? Pour ce TP nous recommandons 256.

4) A quoi correspond le paramètre `blocksize`? Essayez de le changer à `[4, 4]` et à `[16, 16]` et observez les résultats. Quel est l'impact sur la qualité de débruitage? Pour ce TP nous recommandons `[8, 8]`.

5) A quoi correspond le paramètre `stride`? Essayez de le changer à `[2, 2]` : quel est l'impact sur le débruitage et sur le temps de calcul? Pour ce TP, veuillez maintenir `stride=blocksize`.

Partie II : Reconstruction Parcimonieuse

6) Ouvrez la fonction `dictionary_denoise.m` pour comprendre ses entrées, ses sorties, et son fonctionnement général. Quel algorithme évoqué dans le cours implémente la fonction `sparse_coding.m` : (a) Matching Pursuit ou (b) LASSO?

7) A quoi correspond le paramètre `smax`? Comment est-il fixé dans le script principal? Essayer de le fixer à 1 puis zoomez sur l'image débruitée obtenue. Qu'observez-vous et pourquoi?

8) A quoi correspond le paramètre `epsilon`? Comment est-il fixé dans le script principal et pourquoi? Essayez de remplacer le multiplicateur $\sqrt{2}$ par les valeurs suivantes : $\sqrt{0.5}$, $\sqrt{1}$, $\sqrt{4}$. Quel est l'impact sur le nombre d'atomes utilisés par pixel? Sur le temps de calcul? Sur la qualité de débruitage? Pourquoi? Nous recommandons $\sqrt{2}$ pour la suite du TP.

Partie III : Apprentissage de Dictionnaire avec K-SVD

Nous voulons maintenant apprendre un dictionnaire à partir de morceaux de l'image plutôt que d'utiliser le dictionnaire DCT prédéfini.

9) Implémentez l'algorithme d'apprentissage de dictionnaire K-SVD [1] en complétant le fichier `my_ksvd.m` fourni et en vous aidant des diapos de cours.

10) Dans le script principal, ouvrez une nouvelle section `%% Dictionary Learning` avant la section `%% Show results` pour y tester votre implémentation de K-SVD. Commencez par convertir l'image bruitée en un ensemble `Y` de 10,000 morceaux pour entraîner K-SVD en utilisant la fonction `im2blocks` fournie :

```
numblocks = 10000; % Number of training patches
Y = im2blocks(imnoise,numblocks,blocksize);
```

Lancez l'algorithme K-SVD sur `Y` pour 15 itérations, en initialisant avec le dictionnaire DCT. Débruitez ensuite l'image avec le dictionnaire obtenu, de la même façon qu'avec le dictionnaire DCT. Dans la section `%% Show results` affichez l'image ainsi débruitée, le dictionnaire appris, et le nombre d'atomes utilisés par pixel de la même façon que pour la DCT, mais en plaçant les fenêtres au sud pour comparer (`south` et `southeast`).

11) Comparez les résultats obtenus avec votre dictionnaire appris à ceux de la DCT, visuellement et en terme de PSNR, sur les différentes images fournies. Que remarquez-vous ? Examinez les atomes appris par K-SVD : que remarquez vous ? Observez les résultats avec `smax=1`. Pour encore améliorer les résultats, essayez de réduire le paramètre `stride`. Que se passe-t-il en entraînant le dictionnaire sur une image et en testant sur une autre ?

Partie IV : Implémentations Avancées (Bonus)

12) Modifiez la fonction `my_ksvd` pour afficher l'erreur de reconstruction relative $\|Y - DX\|_2^2 / \|Y\|_2^2$ à chaque itération et vérifiez qu'elle décroît bien. Implémentez un critère d'arrêt précoce lorsque cette erreur est inférieure à 0.1%.

13) Vous aurez peut-être remarqué que le premier atome du dictionnaire appris, constant car correspondant à la "fréquence 0" de la DCT initiale, n'est jamais utilisé et n'évolue donc pas. Modifiez `my_ksvd` pour remplacer tout atome inutilisé par le signal le moins bien reconstruit de `Y` dans l'itération en cours.

14) Réfléchissez à une façon de modifier le code fourni pour faire de l'*inpainting* plutôt que du débruitage, c'est-à-dire, pour "repeindre" des pixels complètement manquant dans une image non bruitée. On pourra par exemple considérer le cas où une colonne de pixels est entièrement manquante.

Références

[1] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD : An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.