

Module CETS8AH

TP Détection d'indices

Séance du 22 mars 2022.

Le compte rendu de ce TP sous forme word ou pdf est à rendre pour le **5 avril** en le déposant sur arche. Ce compte rendu doit contenir les réponses **argumentées** aux questions posées ainsi que des images illustrant ces résultats. Vous joindrez à ce compte rendu une archive contenant les fichiers .m implémentant les fonctions demandées. Histoire d'alléger le trafic, ne joignez pas à votre zip les données du TD. Je les ai déjà!

Le cours, le texte du TD et les données images à utiliser sont disponibles sur arche et sur ma [page web](#). Nous utiliserons aussi dans ce TD le réseau de neurones AlexNet.

1 Récupérer le réseau Alexnet

AlexNet est ainsi nommé car son inventeur est Alex Krizhevsky et ses collègues. Il a été décrit dans l'article *ImageNet Classification with Deep Convolutional Neural Networks*, publié à la conférence NIPS en 2012 (Neural Information Processing Systems), que vous pouvez retrouver [ici](#). C'est un réseau très connu car ce fut le premier réseau, implémenté sur GPU, à gagner (de beaucoup) une compétition de reconnaissance automatique d'images.

Ce réseau peut être chargé dans MATLAB avec les poids appris sur la base ImageNet ([Imagenet](#) est une très grosse base d'images conçue pour l'apprentissage et la reconnaissance d'objets). Pour cela, il suffit d'entrer la commande suivante qui le charge dans la variable net:

```
» net = alexnet;
```

Il est possible (probable ?) que vous aurez un message d'erreur : Error using alexnet (line 51) alexnet requires the Deep Learning Toolbox Model for AlexNet Network support package. To install this support package, use the Add-On Explorer. Si c'est le cas, cliquez sur Add-On Explorer . Une fenêtre s'ouvre : cliquez sur le bouton "Install" à droite. Il vous est alors demandé de vous connecter avec votre compte MathWorks. Si vous n'en avez pas, cliquez sur "Create Account", et renseignez votre mail étudiant, votre pays, nom et prénom, puis cliquez sur "Continue". Allez alors voir votre boîte mail et cliquez sur le lien de confirmation, puis revenez à la fenêtre de création de compte, cliquez sur "Continue", puis indiquez un mot de passe, et cochez la case "I accept the Online Services Agreement". Cliquez sur "Continue". L'installation d'AlexNet se lance automatiquement, et démarre dès que vous cliquez sur "Accept".

2 Descripteurs issus des réseaux de neurones

L'objectif de cet exercice est d'utiliser des descripteurs globaux des images pour retrouver, étant donnée une image test, l'image qui lui est la plus proche dans une base de donnée. C'est ce qu'on appelle "image retrieval" en anglais. On s'intéresse ici à l'usage de descripteurs *convnet* issus des réseaux convolutionnels (CNN). L'idée générale est d'utiliser des systèmes de classification appris sur de grosses bases de données avec un grand nombre de classes, et de considérer comme descripteur la carte d'indice précédent l'étape de classification. Le réseau ayant été appris pour classifier, il est ainsi fort probable que cette dernière carte a un fort pouvoir discriminant et peut être utilisée pour mesurer la ressemblance entre deux images.

On considère ici le réseau alexnet, que vous chargerez avec la commande `net=alexnet`. Visualisez les couches de ce réseau par la commande `net.Layers`. On va retenir ici comme descripteur la couche `fc7` qui précède la classification.

Les données nécessaires à cet exercice se trouvent dans le répertoire UsineReducit de l'archive. Il contient des images de diverses machines d'une usine, prises à des endroits et des moments différents. Ces images sont à la taille [227,227] et donc directement utilisables par le réseau AlexNet.

- regardez dans le help de matlab la page consacrée à *visualize activation of a convolutional neural network*. Celle ci vous permet de voir, pour une image donnée *im* en entrée, toutes les cartes correspondant à cette image obtenues à divers couches du réseau. La façon d'obtenir le résultat du réseau pour la couche *fc7* est donc (*activ= activations(net,im, 'fc7')*) Quelle est la taille de cette activation? Pour éviter les problèmes de représentation, vous en ferez un tableau par la commande *reshape(activ,[1,4096])*, que nous appellerons descripteurs dans la suite.
- étant donnée une image *im*, on veut trouver les images les plus ressemblantes se trouvant dans le répertoire *UsineReduit*. Pour cela vous allez calculer le descripteur de chacune des images de ce repertoire. Pour trouver les images les plus semblables, on va comparer le descripteur de l'image test à chacun des descripteurs des images de la base. On utilise pour cela le cosinus entre les deux vecteurs de descripteurs, qui peut se calculer via le produit scalaire de ces deux vecteurs, ces vecteurs ayant été préalablement normalisés. Expliquer pourquoi cette métrique est adaptée pour mesurer la ressemblance.
- Ecrire une fonction *plusProchesImages* qui prend en entrée le nom d'une image et un entier *N* et sort les noms des *N* images les plus semblables au sens de la métrique précédemment définie.¹
- Tester le résultat de votre méthode en prenant pour entrée *usine6.JPG* *usine8.JPG* et *usine40.JPG* qui sont des images du répertoire *UsineReduit*. Afficher dans chacun des cas les 9 images les plus proches (vous pouvez par exemple utiliser la fonction *imtile* pour les afficher dans une représentation avec 3 ligne et 3 colonnes mais c'est optionnel).
- analyser la pertinence des résultats (vous pouvez bien sur tester sur plus que les trois images proposées). Comment jugez vous la qualité des résultats? Dans chacun des cas, combien d'images, parmi les 9, correspondent bien à l'image requête? Pensez-vous qu'on pourrait facilement sélectionner les images qui correspondent bien à la scène grâce à un seuil sur la valeur du descripteur?
- nous avons pris la couche *fc7* du réseau alexnet, qui fournit un descripteur à 4096 éléments. Nous aurions aussi pu prendre la couche *fc8* qui fournit un descripteur à 1000 éléments. Reprenez donc maintenant la recherche des images les plus proches avec ce descripteur à 1000 éléments. Comparez les résultats des deux expériences obtenus avec les couches *fc7* et *fc8*.

3 Quelques affichages

On souhaite ici regarder comment afficher des points ou des lignes sur une image, ce qui nous servira plus tard pour afficher des indices sur les images.

On considère une liste de points à afficher, *tabx* contenant les coordonnées en x et *taby* celles en y *tabx = [100 200 330 150 200]* *taby = [200 350 400 500 100]* L'appel *plot(tabx,taby)* affiche les points 2d, en le reliant par des segments de droites. En utilisant la documentation de la fonction *plot*:

- faites varier l'épaisseur de la ligne avec l'option *'LineWidth'*.
- afficher ces 5 points représentés par une croix, sans ligne entre eux.
- utiliser la fonction *hold* pour afficher ces points sur une image de votre choix avec la suite de commandes: *i1=imread('l'image de votre choix');* *hold on;* *plot(tabx,taby)*.

Comparer avec l'affichage précédent des mêmes points, sans image de fond. Vous constaterez que l'axe des x utilisées suite à l'affichage une image est l'axe horizontal. Pour afficher correctement vos points sur l'image, vous devrez donc faire *plot(taby,tabx)*.

¹Pour lister les images d'un repertoire, utiliser par exemple *liste=dir('monRepertoire')*, les noms des images sont alors contenues dans *liste(i).name*.



a.



b.

Figure 1: a: le point à mettre en correspondance dans I_1 , b: le point avec la corrélation maximum dans I_2 pour $w=10$

4 Mise en correspondance par corrélation

Le but de cet exercice est de comprendre un peu mieux les méthodes de mise en correspondance par corrélation. Il existe dans matlab la fonction `normxcorr2` qui effectue une corrélation croisée normalisée entre un template (i.e. une matrice représentant un motif) et chaque point de l' image (en considérant une fenetre de même taille que le template autour de ce point). Cette fonction identifie dans la carte de corrélation la position du template ayant la plus forte ressemblance (voir la doc de cette fonction dans Matlab), au sens du produit scalaire.

En notant le template par t et l'image par f , a corrélation utilisée est la corrélation centrée normalisée définie au point (i, j) de l'image par:

$$r_{ij} = \frac{\sum_{m=-w}^w \sum_{n=-w}^w (f(m+i, n+j) - \bar{f})(t(m, n) - \bar{t})}{\sqrt{(\sum_m \sum_n (f(m+i, n+j) - \bar{f})^2)(\sum_m \sum_n (t(m, n) - \bar{t})^2)}}$$

où \bar{f} et \bar{t} sont les valeurs moyennes de t et f sur la fenetre considérée

Nous allons utiliser cette fonction pour faire de la mise en correspondance de points, c'est-à-dire identifier, pour un point donné d'une image I_1 , le point lui ressemblant le plus dans l'image I_2 . Nous allons considérer le point de coordonnées $m_1 = [108, 233]$, les coordonnées étant exprimées dans la représentation matricielle de cette image. Ce point est intéressant (car difficile à mettre en correspondance) car il s'agit d'un motif répétitif de la structure que nous allons étudier (fig.1.a).

Nous allons considérer pour image I_1 l'image `castle.018Red.jpg` présente dans l'archive. Considérons pour seconde image `castle.010Red.jpg`.

1. Vous allez considérer une fenêtre de taille w autour de m_1 , c'est à dire le template $I(m_1(1) - w : m_1(1) + w, m_1(2) - w, m_1(2) + w)$. Consultez la doc de la fonction `normxcorr2`. Appliquez-là entre `template` et I_2 afin de trouver la fenêtre de demie-taille w dans I_2 ressemblant le plus à `template`. Le centre de cette fenêtre sera alors considéré comme le correspondant de m_1 . Affichez la fenêtre de corrélation sur l'image cible, comme je l'ai fait dans la figure 1. Vous noterez que vous pouvez visualiser la carte de corrélation, et que la taille de cette carte est égale à la dimension de l'image initiale augmentée de $2w$. La commande `[ypeak, xpeak] = find(c == max(c(:)))` utilisée dans la doc donne les coordonnées réalisant le maximum de la carte de corrélation dans c et correspondent donc dans l'image initiale à `ypeak - w` et `xpeak - w`.
2. On souhaite maintenant examiner non plus le point réalisant le maximum de la carte de corrélation mais les N points fournissant les scores les plus élevés. Ecrivez une fonction prenant en paramètres

N et la demie taille de fenêtre w et fournissant en sortie les N points de I_2 ayant la corrélation la plus grande. Affichez ces points en identifiant le point de corrélation maximale avec une couleur différente.

- Tester votre fonction avec les différentes tailles de fenêtres $w=3, 5, 15, 25$ et $N=25$.
- Quelle est l'évolution de la carte de corrélation avec la croissance de w ?
- Expliquer pourquoi la valeur maximale ne correspond pas au point correspondant pour $w=3$. Pourquoi les N points avec la plus grande corrélation sont ils éparpillés quand w est petit et concentrés en général au même point quand w croit?
- vous aurez constaté qu'il y a parfois des points qui se promènent dans le ciel et induisent une corrélation forte... Pouvez vous expliquer pourquoi?