Transfer Learning par extracion de feature dans un CNN

Loüet Joseph & Karmim Yannis
Spécialité **DAC**

Table des matières

1 Architecture VGG16 **3**

2 Transfer Learning avec VGG16 sur 15 Scene **5**
2.1 Principe de la démarche 5
2.2 Extraction des features de VGG16 5
2.3 Apprentissage de classifieurs SVM. 5

1 Architecture VGG16

Questions

1) La dernière couche de max-pooling renvoie une sortie de dimension $7 \times 7 \times 512 = 25\,088$. La première couche de fully-connected prend en entrée la sortie du max-pooling et renvoie une sortie de dimension 4096. Ainsi, la première couche de fully-connected doit apprendre $(25088 + 1) \times 4096 = 102\,764\,544$ poids. De même, la seconde couche de fully-connected renvoie une vecteur de taille 1000 et prend en entrée un vecteur de taille 4096. Ainsi, la deuxième couche de fully-connected doit apprendre $(4096 + 1) \times 1000 = 4\,097\,000$ poids.
De ce fait, le réseau VGG16 doit apprendre au moins 106861544 poids.

2) La sortie de la dernière couche de ce réseau est un vecteur de taille 1000. Cela correspond aux 1000 classes de la base d'images ImageNet. Nous appliquons un softmax sur la dernière couche de fully-connected pour obtenir une distribution de probabilité sur les 1000 classes.

3) Si nous essayons de prédire l'image suivante :

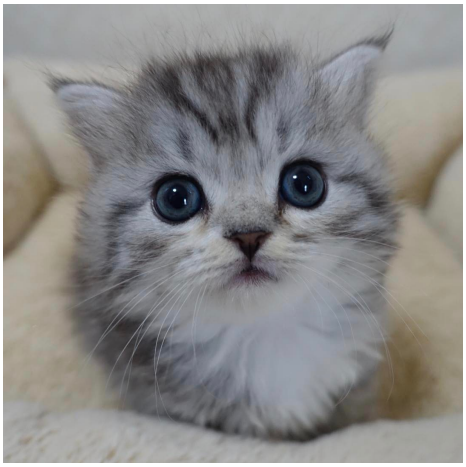

Figure 1 – Image de chat mignon

Le réseau VGG16 nous prédit la classe 'Persian cat' avec une probabilité de 0,970 (à l'aide d'un softmax sur le résultat du réseau).

Le réseau reconnait bien nos vêtements sur la figure 2, en effet il remarque le cardigan de Joseph (à gauche) et le pull en laine/fourrure de Yannis (à droite).

4) On remarque sur la carte de la figure 4 le museau du chat de la Figure 1 ainsi que ses yeux.

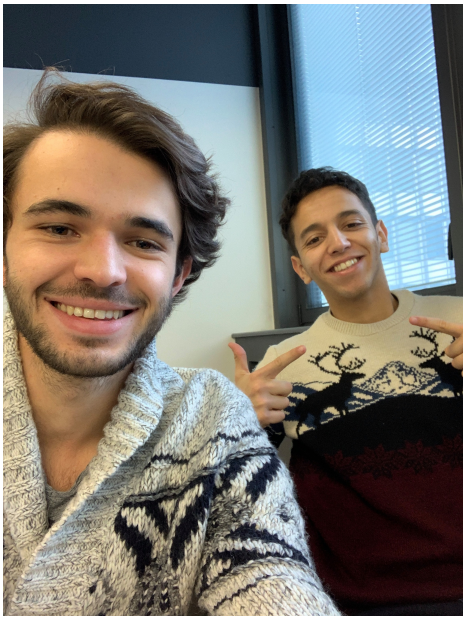

FIGURE 2 – Image de binome mignon

```
VGG16 reconnait:  ['stole', 'fur coat', 'poncho', 'cardigan', 'bath towel']  sur l'image donnée, avec d
s probabilités respectives de : [0.5275835, 0.20363484, 0.06483848, 0.059744027, 0.015212317]
```

FIGURE 3 – Résultat de la prédiction du modèle VGG16 pour la figure 2

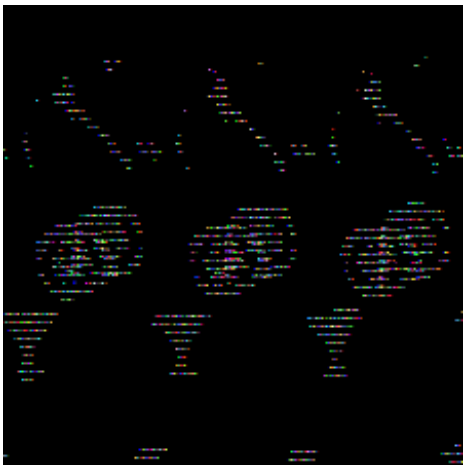

FIGURE 4 – Une des 64 cartes obtenus après le passage de la première couche de convolution sur l'image de la figure 1.

2 Transfer Learning avec VGG16 sur 15 Scene

2.1 Principe de la démarche

Questions

5) Nous apprenons pas directement VGG16 sur 15 Scene car le réseau n'est pas adapté pour les prédictions que nous souhaitons faire. En effet, la dernière couche renvoie un vecteur de taille 1000 correspondant aux 1000 classes du dataset ImageNet.

6) Le pré-apprentissage sur ImageNet peut aider à la classification car les premières couches de cet apprentissage permet de détecter les particularités de l'image (courbures, points, cercles, etc...). C'est pour cela que nous ne modifierons pas les poids des premières couches de ce réseau mais seulement la dernière couche permettant la classification.

7) Les limites de cet apprentissage sont que l'apprentissage de ce réseau s'est effectué sur des images photographiées d'une certaine manière et le réseau s'est entraîné sur les particularités de ces images. De ce fait, si les images que nous allons classifier sont différentes que celles d'ImageNet et le réseau aura plus de difficultés les particularités des nouvelles images puisqu'il sera pré-entrainé sur des images avec d'autres caractéristiques.

2.2 Extraction des features de VGG16

Questions

8) La couche à laquelle les features sont extraites correspond à la couche de classification pour les 1000 classes d'ImageNet.

9) Les images sont codés en noir et blanc, de ce fait, les images sont codés sur une seule 'couche' alors que le codage RGB est sur 3 'couches'. Pour remédier à ce problème, nous allons copier la seule 'couche' du codage en noir et blanc et la recopier 3 fois pour obtenir une image en noir et blanc. De plus, pour que le réseau VGG16 soit efficace sur nos images, nous allons normalisé nos images selon la moyenne et la variance du dataset ImageNet.

2.3 Apprentissage de classifieurs SVM.

10) On obtient un score accuracy sur le SVM de **0.89**, alors que nous avons obtenu un score accuracy de **0.67** avec la technique Bag Of Words. Les réseaux de convolutions sont donc bien plus efficace pour l'extraction de nos features.

11) On peut à la place d'utiliser un SVM pour classifier nos features extrait du réseau pré-entraîné VGG16 utiliser un réseau de neurones fully-connected avec en sortie les 15 classes de notre problème. On peut donc reprendre le code de VGG16 en modifiant les deux dernières couches par exemple, par un réseau fully-connected. Cependant il faut être prudent lors de l'apprentissage, puisque le backward peut modifier les poids des premières couches de notre réseau pré-entraîné. On peut donc utiliser les fonctionnalités *freeze* de pytorch pour ne pas modifier les poids du début de notre réseau.

12)

• Le paramètre C du SVM n'influe pas vraiment sur les scores pour notre tâche de classification 15scene. Pour différentes valeurs de C [1e-3, 1e-2, 1e-1, 1, 10, 100, 1000] on a un score qui varie de **0.89** à **0.894**.