

le Transfer Learning

Ibrahim & Youenn


P3 ISEN Simplon

Qu'est ce que l'Architecture VGG16 ?

L'architecture **VGG16** est un réseau de neurones convolutifs (ConvNets) profond qui a été développé par le **Visual Geometry Group (VGG)** de l'**Université d'Oxford**. Il est composé de **16 couches de convolution et de pooling**, d'où son nom.

Ce réseau neuronal convolutif a été formé sur la base de données **ImageNet**, qui contient plus d'un million d'images classées dans 1000 catégories.

VGG16 a obtenu des résultats exceptionnels lors de sa première présentation, obtenant une précision de classification de **92,7%** sur ImageNet. Il est depuis devenu l'une des architectures de CNN les plus populaires, et est utilisé dans de nombreux domaines, tels que la reconnaissance faciale, la détection d'objets et la classification d'images.



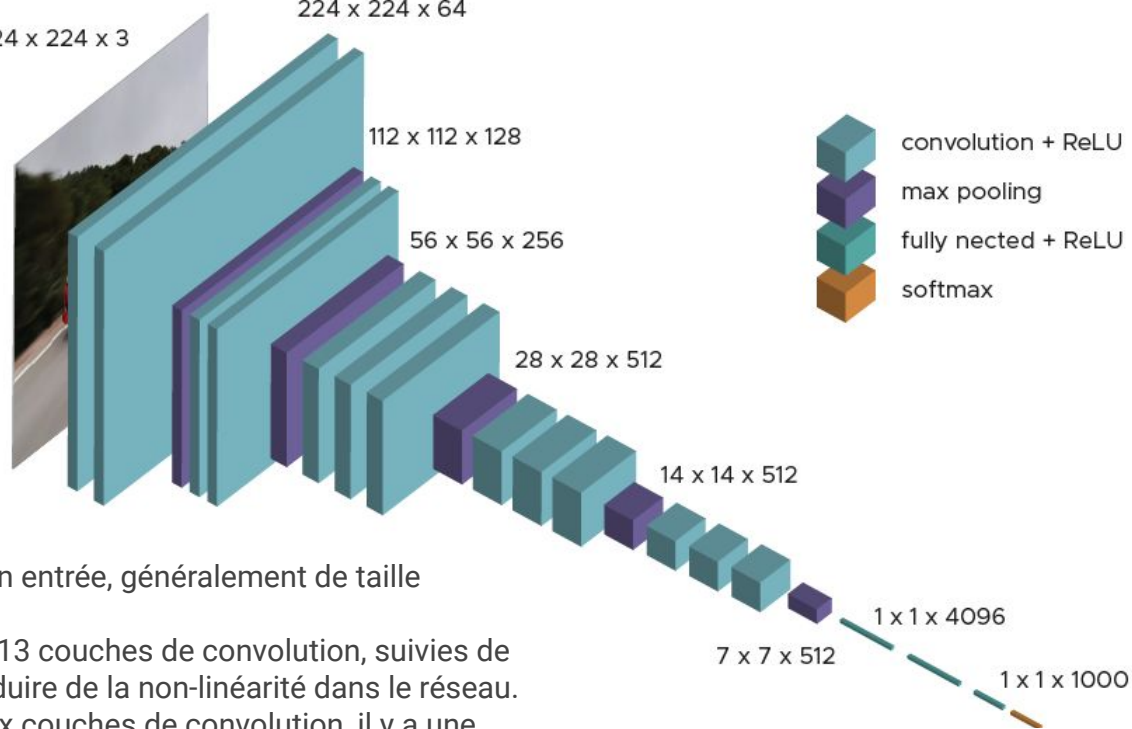
Architecture VGG16

Les **16 couches** de **VGG16** sont organisées en 5 blocs, chacun contenant 2 ou 3 couches convolutives.

Les blocs sont séparés par des couches de pooling, qui servent à réduire la taille des images tout en conservant les caractéristiques importantes.

Chaque couche convolutive utilise des filtres de 3.3

Les filtres ne sont pas prédéfinis, elle font parti des parametre que le modèle apprend de lui même. il peut ainsi apprendre à détecter des caractéristiques spécifiques dans les images; bords, coins, formes géométriques.



- **Couche d'entrée** : Cette couche reçoit une image en entrée, généralement de taille 224x224x3 (pour une image en couleur RGB).
- **Couches de convolution** : VGG16 est constitué de 13 couches de convolution, suivies de couches de ReLU (Rectified Linear Unit) pour introduire de la non-linéarité dans le réseau.
- **Couches de pooling** : Après chaque groupe de deux couches de convolution, il y a une couche de pooling max-pooling qui réduit la dimension spatiale de la sortie.
- **Couches entièrement connectées** : Après les couches de convolution et de pooling, il y a trois couches entièrement connectées avec des couches de ReLU.
- **Couche de sortie** : La couche de sortie a généralement 1000 neurones pour la classification des images en 1000 catégories différentes, comme dans le jeu de données ImageNet

Chargement de VGG depuis Keras

Chargement d'un modèle VGG16 sous Keras sans transfer learning :

```
from keras.applications import VGG16
from keras.models import Sequential

# Créer un modèle Sequential
model = Sequential()

# Charger le modèle VGG16 sans les poids pré-entraînés
vgg16_model = VGG16(weights=None, include_top=False, input_shape=(224, 224, 3))

# Ajouter les couches du modèle VGG16 au modèle Sequential
model.add(vgg16_model)

# Ajouter votre propre couche de sortie adaptée à votre tâche
model.add(...) # Ajoutez vos couches ici

# Compiler le modèle
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Résumé du modèle
model.summary()
```

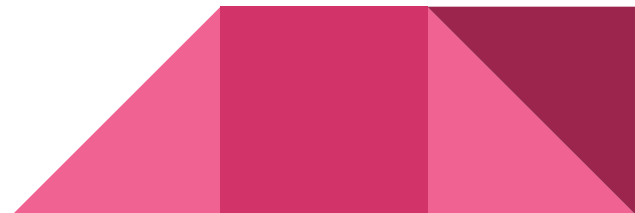
Chargement de VGG depuis Keras

Chargement d'un modèle VGG16 sous Keras avec transfer learning :

```
from keras.applications import VGG16

# Charger le modèle VGG16 pré-entraîné
vgg16_model = VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
```

Dans cet exemple, `weights='imagenet'` signifie que nous chargeons les poids pré-entraînés sur ImageNet. `include_top=False` signifie que nous excluons la couche de sortie du modèle car nous allons ajouter notre propre couche de sortie adaptée à notre tâche.



Transfer Learning : définition et modèles pré-entraînés sur ImageNet

Plutôt que de former un modèle à partir de zéro pour une tâche particulière, le transfer learning permet de prendre un modèle déjà entraîné sur une tâche générale et de le réutiliser comme point de départ pour une nouvelle tâche.

