

Deep Learning pour la vision par ordinateur

TD_Transfer Learning

1. ResNet50

- Etudier, prendre en main et exécuter le Tutorial Keras: Transfer Learning with ResNet50 for image classification on Cats & Dogs dataset, <https://www.kaggle.com/code/suniliitb96/tutorial-keras-transfer-learning-with-resnet50>
- Adapter ce tutoriel à la problématique de classification CNN de la base COVID-19, utilisée au Lab précédent ;
- Comparer les résultats obtenus avec la meilleure architecture que vous avez conçue au Lab 2, conclure,
- Améliorer les résultats de classification et justifier vos choix

2. VGG-19

- Même travail que le 1. Avec une autre architecture (VGG19), exemple de code ci-dessous ; Comparer les résultats obtenus avec ceux de ResNet50 ; Améliorer les résultats de classification et justifier vos choix

```
from tensorflow.keras import applications
from keras.models import Sequential
from keras.layers import Dropout,Dense,Conv2D,GlobalAveragePooling2D
from keras.models import Model
from tensorflow.keras.optimizers import Adam
img_width,img_height=224,224
base_model
=applications.VGG19(weights='imagenet',include_top=False,input_shape=(img_width,img_height,3))
top_model=Sequential()
top_model.add(GlobalAveragePooling2D(input_shape=base_model.output_shape[1:],data_format=None))
top_model.add(Dense(256,activation='relu'))
top_model.add(Dropout(0.25))
top_model.add(Dense(1,activation='sigmoid'))
model=Model(inputs=base_model.input,outputs=top_model(base_model.output))
```

Rq. Analyser à chaque fois l'architecture utilisée et commenter la relation avec les résultats obtenus.