Projet de classification binaire par Réseau neuronal convolutif

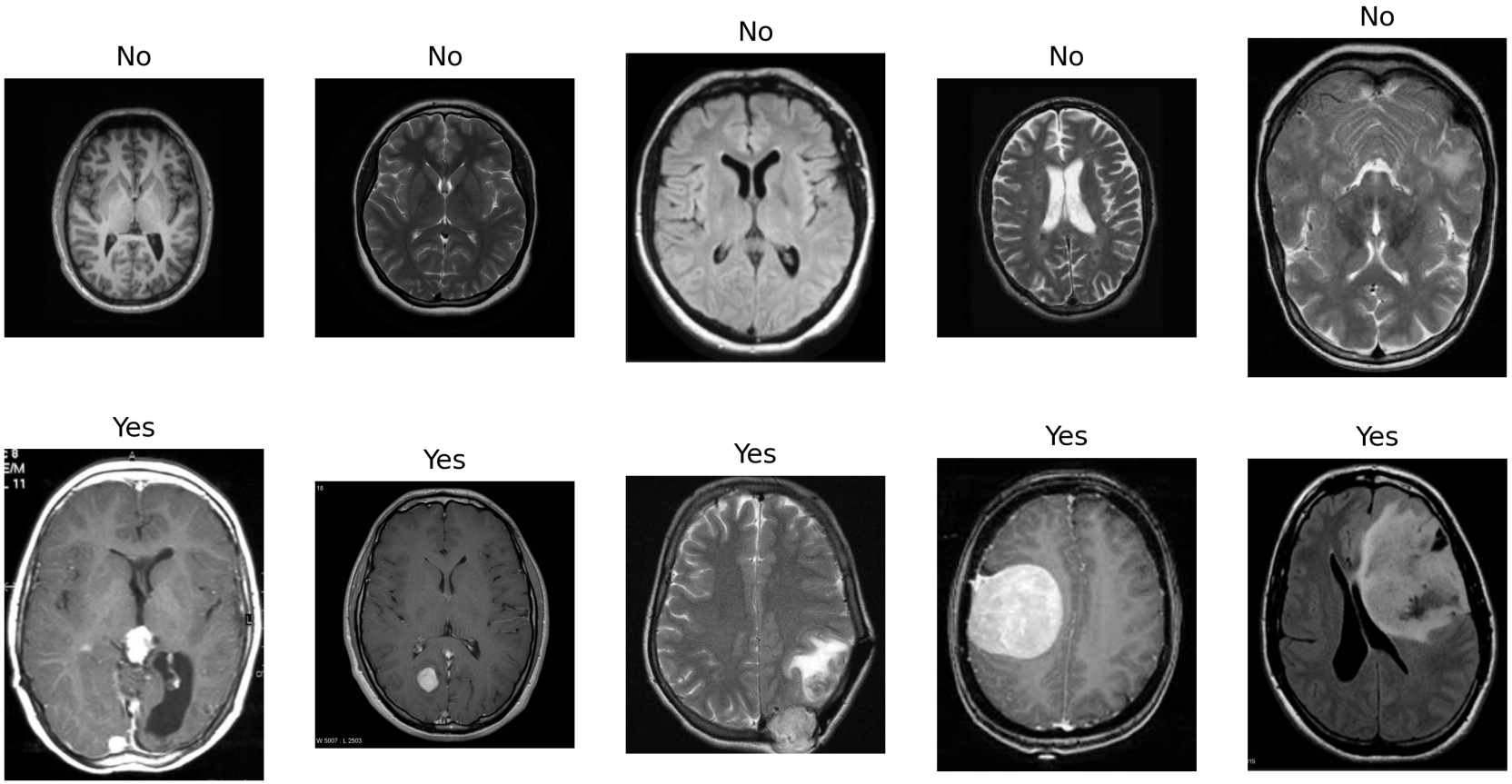
Application à l'imagerie médicale

LARIBI Yanis

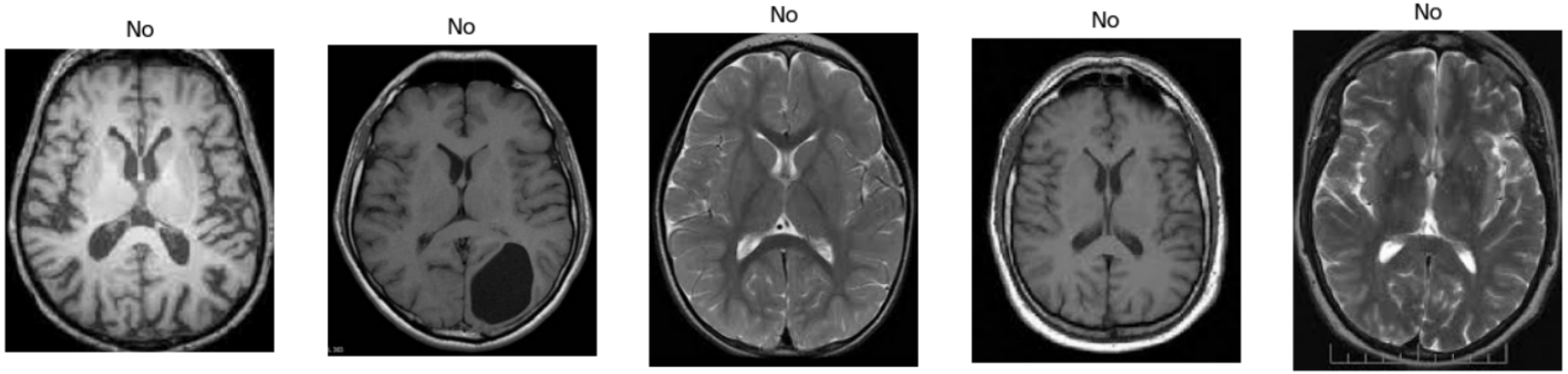
École des Mines de Nancy

Sommaire

Exemples d'images du dataset

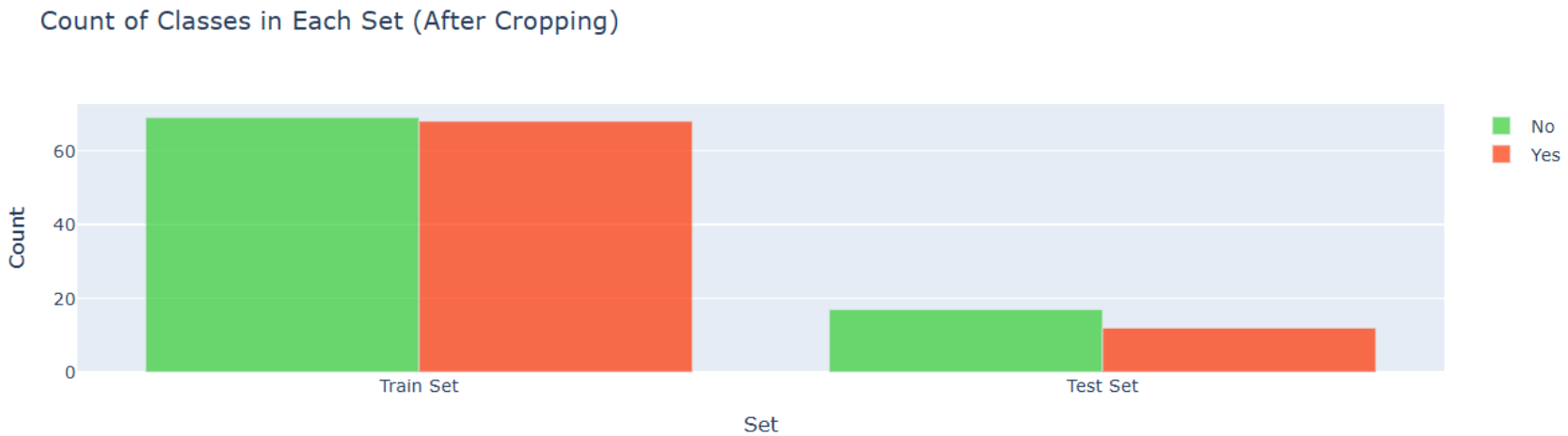


Exemples d'images recadrées (No)

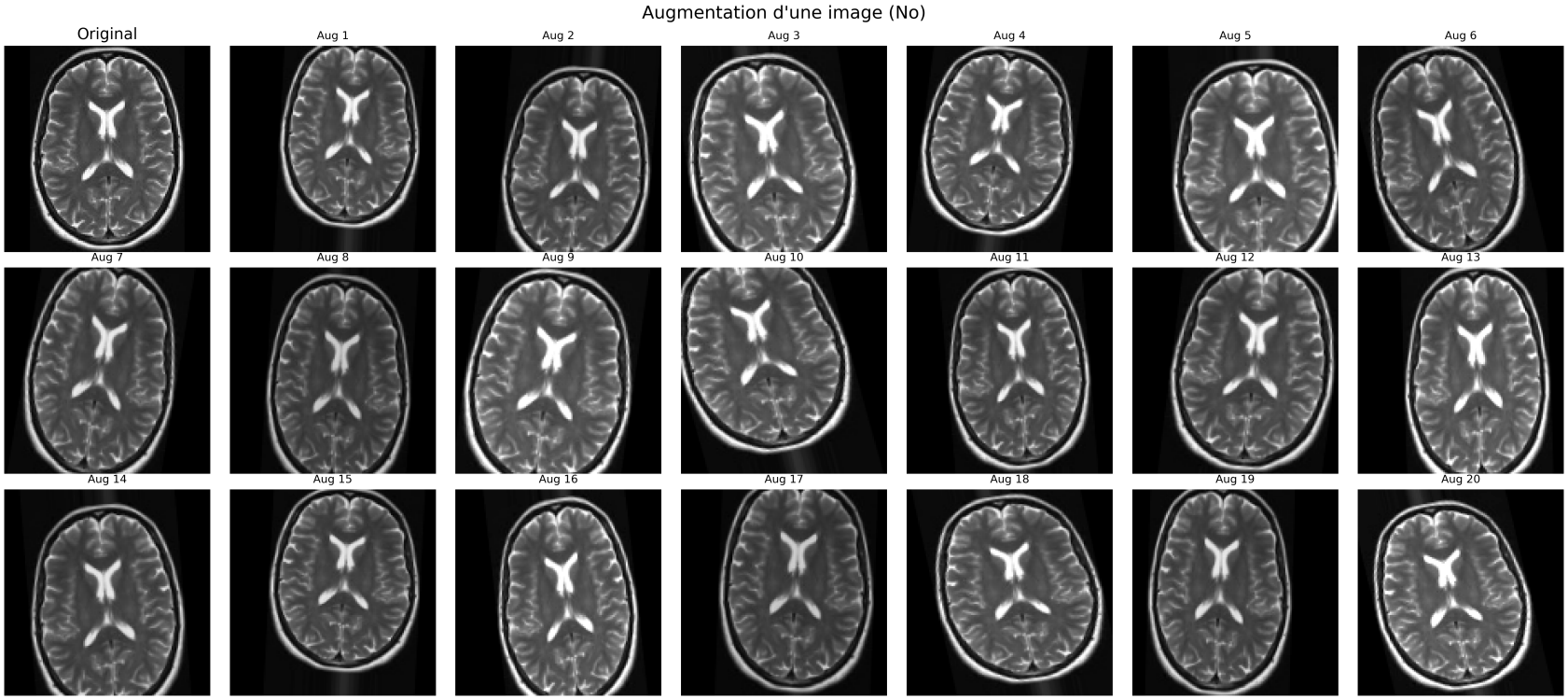


Exemples d'images recadrées (Yes)

Répartition des données train/test ≈ 80/20



- 137 images d'entraînement : 69 No et 68 Yes
- 29 images de test : 17 No et 12 Yes
- Faible nombre d'images d'entraînement ⇒ Augmentation de données.



- Opérations effectuées : Rotation, zoom, élargissement, modification du contraste, horizontal flip, brightness.

Construction des modèles CNN

Layer (type)	Output Shape	Param #
conv2d_20 (Conv2D)	(None, 128, 128, 32)	896
leaky_re_lu_20 (LeakyReLU)	(None, 128, 128, 32)	0
max_pooling2d_19 (MaxPooling2D)	(None, 64, 64, 32)	0
batch_normalization_19 (BatchNormalization)	(None, 64, 64, 32)	128
conv2d_21 (Conv2D)	(None, 64, 64, 64)	18,496
leaky_re_lu_21 (LeakyReLU)	(None, 64, 64, 64)	0
max_pooling2d_20 (MaxPooling2D)	(None, 32, 32, 64)	0
batch_normalization_20 (BatchNormalization)	(None, 32, 32, 64)	256
conv2d_22 (Conv2D)	(None, 32, 32, 128)	73,856
leaky_re_lu_22 (LeakyReLU)	(None, 32, 32, 128)	0
max_pooling2d_21 (MaxPooling2D)	(None, 16, 16, 128)	0
batch_normalization_21 (BatchNormalization)	(None, 16, 16, 128)	512
conv2d_23 (Conv2D)	(None, 16, 16, 256)	295,168
leaky_re_lu_23 (LeakyReLU)	(None, 16, 16, 256)	0
max_pooling2d_22 (MaxPooling2D)	(None, 8, 8, 256)	0
batch_normalization_22 (BatchNormalization)	(None, 8, 8, 256)	1,024
flatten_5 (Flatten)	(None, 16384)	0
dense_15 (Dense)	(None, 128)	2,097,280
leaky_re_lu_24 (LeakyReLU)	(None, 128)	0
dropout_5 (Dropout)	(None, 128)	0
dense_16 (Dense)	(None, 64)	8,256
leaky_re_lu_25 (LeakyReLU)	(None, 64)	0
dense_17 (Dense)	(None, 1)	65

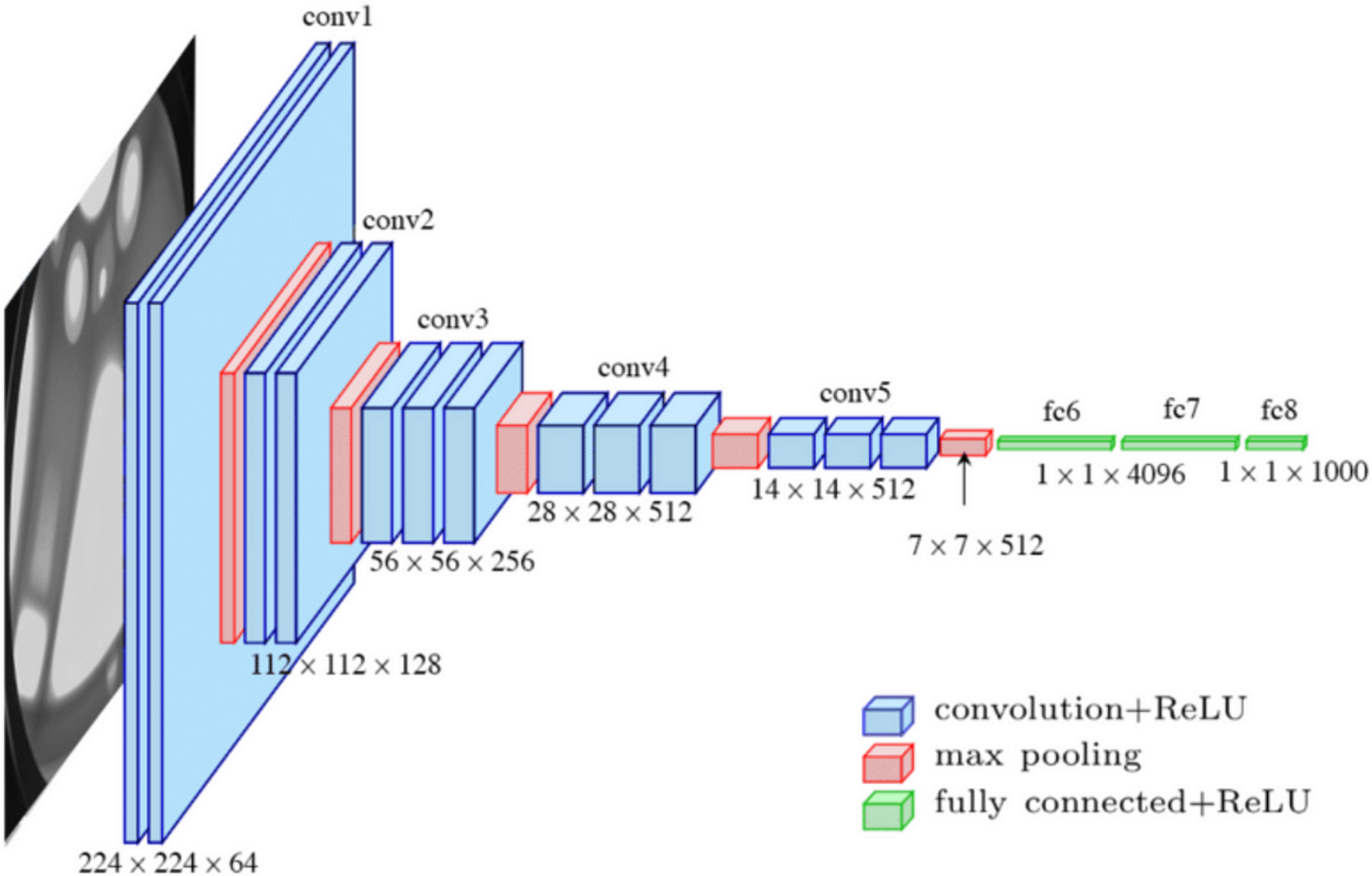
Total params: 2,495,937 (9.52 MB)
Trainable params: 2,494,977 (9.52 MB)
Non-trainable params: 960 (3.75 KB)

Figure: From scratch

Layer (type)	Output Shape	Param #
input_layer_10 (InputLayer)	(None, 128, 128, 3)	0
block1_conv1 (Conv2D)	(None, 128, 128, 64)	1,792
block1_conv2 (Conv2D)	(None, 128, 128, 64)	36,928
block1_pool (MaxPooling2D)	(None, 64, 64, 64)	0
block2_conv1 (Conv2D)	(None, 64, 64, 128)	73,856
block2_conv2 (Conv2D)	(None, 64, 64, 128)	147,584
block2_pool (MaxPooling2D)	(None, 32, 32, 128)	0
block3_conv1 (Conv2D)	(None, 32, 32, 256)	295,168
block3_conv2 (Conv2D)	(None, 32, 32, 256)	590,080
block3_conv3 (Conv2D)	(None, 32, 32, 256)	590,080
block3_pool (MaxPooling2D)	(None, 16, 16, 256)	0
block4_conv1 (Conv2D)	(None, 16, 16, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 16, 16, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 16, 16, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 8, 8, 512)	0
block5_conv1 (Conv2D)	(None, 8, 8, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 8, 8, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 8, 8, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0
global_average_pooling2d_2 (GlobalAveragePooling2D)	(None, 512)	0
dense_27 (Dense)	(None, 256)	131,328
dropout_9 (Dropout)	(None, 256)	0
dense_28 (Dense)	(None, 128)	32,896
dense_29 (Dense)	(None, 1)	129

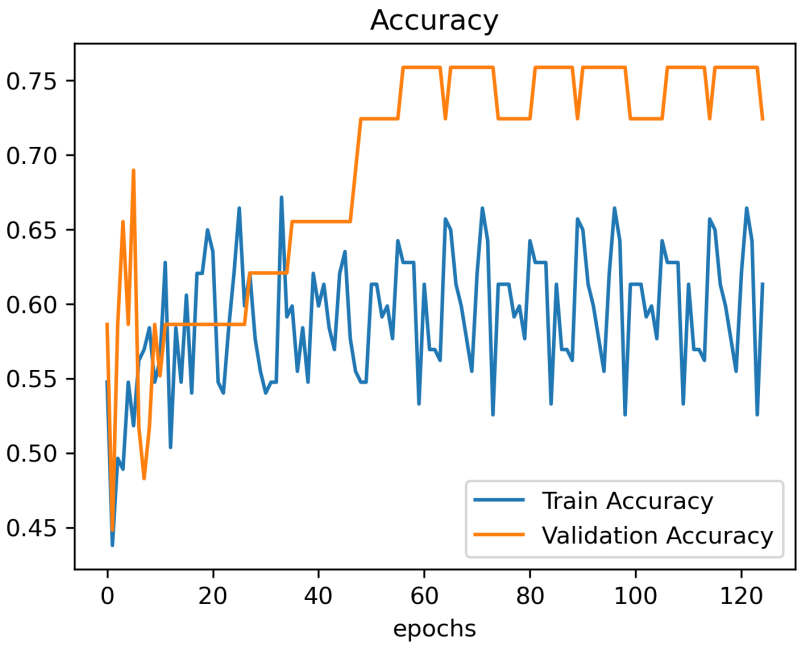
Total params: 14,879,041 (56.76 MB)
Trainable params: 164,353 (642.00 KB)
Non-trainable params: 14,714,688 (56.13 MB)

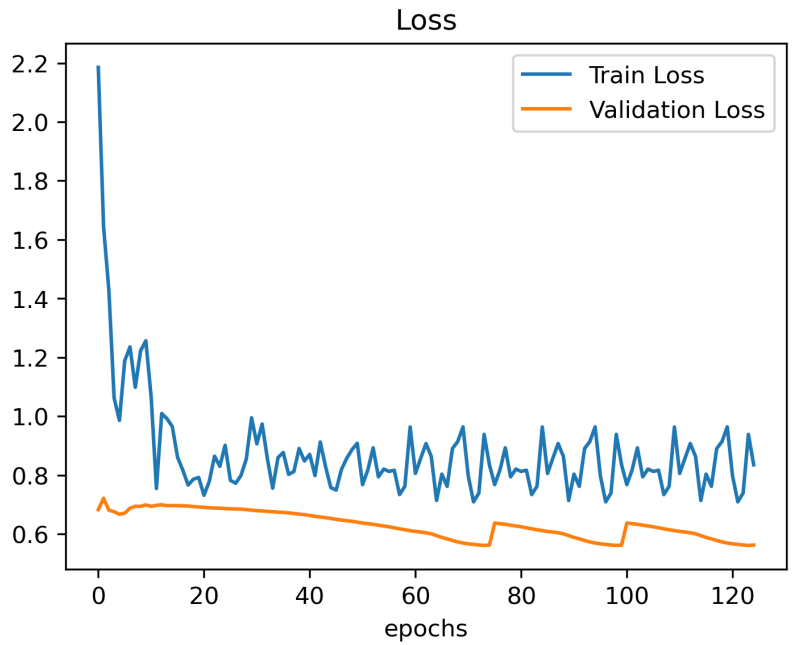
Figure: pré-entraîné sur VGG-16



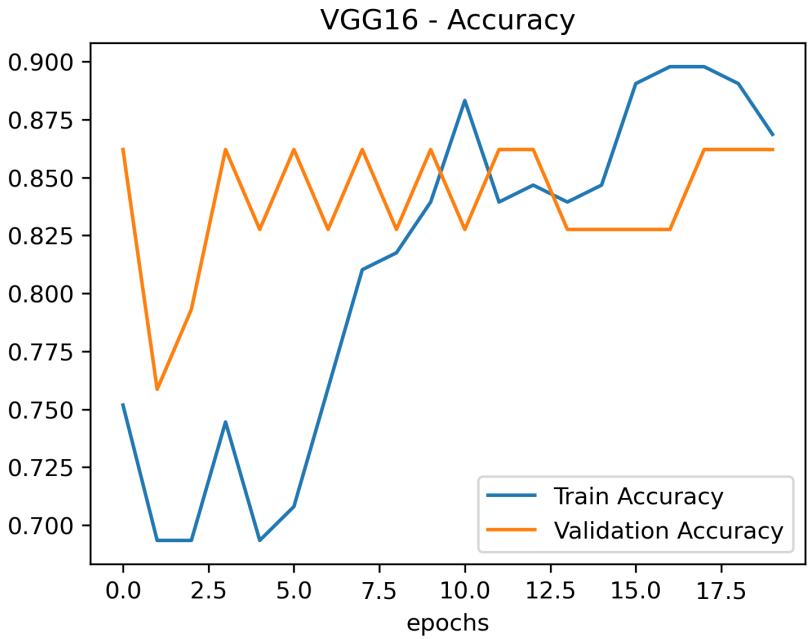
- Architecture de VGG-16 que l'on a fine tuné en rajoutant et entraînant quelques couches supplémentaires.

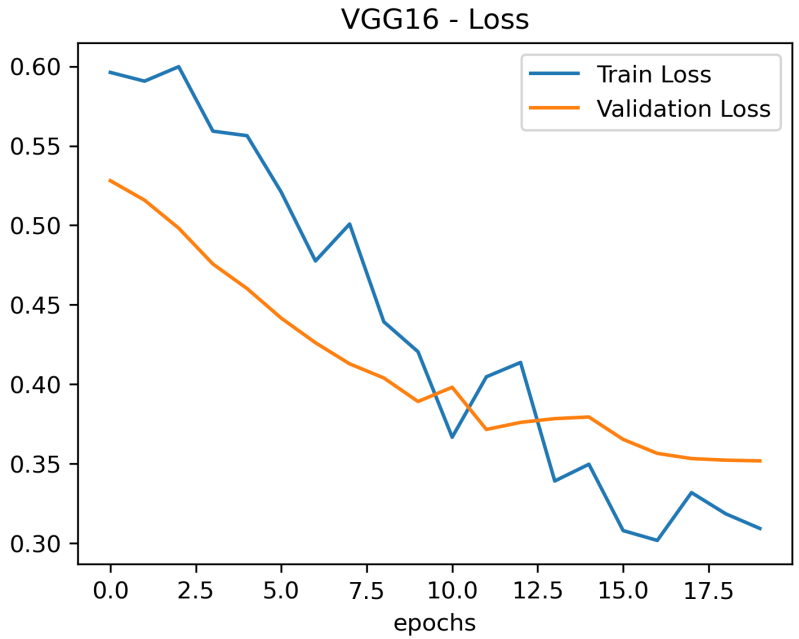
Tracé des courbes pour le CNN scratch





Tracé des courbes pour VGG-16 fine tuné





Test accuracy : comparaison

Comparer les Performances

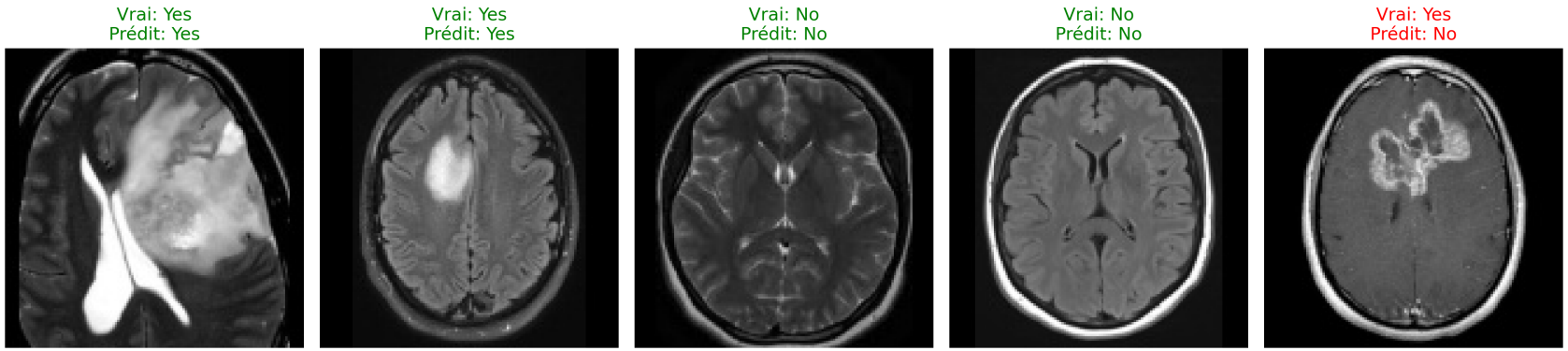
[270]:
```
# Évaluation du modèle entraîné from scratch
test_loss_cnn, test_accuracy_cnn = model.evaluate(test_generator)
print(f"📊 CNN from scratch - Test Accuracy: {test_accuracy_cnn:.4f}")

# Évaluation du modèle pré-entraîné VGG16
test_loss_vgg, test_accuracy_vgg = model_vgg.evaluate(test_generator)
print(f"📊 VGG16 Transfer Learning - Test Accuracy: {test_accuracy_vgg:.4f}")
```

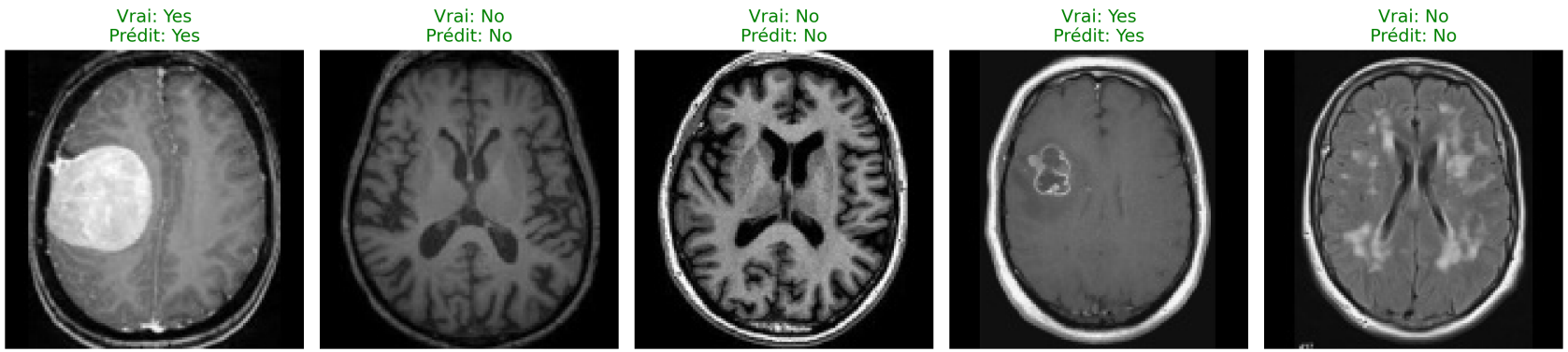
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.7241 - loss: 0.5624
📊 CNN from scratch - Test Accuracy: 0.7241
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 868ms/step - accuracy: 0.8621 - loss: 0.3517
📊 VGG16 Transfer Learning - Test Accuracy: 0.8621
```

Quelques prédictions sur la base de test

Résultats du CNN from scratch - Prédictions vs. Vraies Classes



Résultats du VGG16 fine tuné - Prédictions vs. Vraies Classes



Matrices de confusion sur la base de test

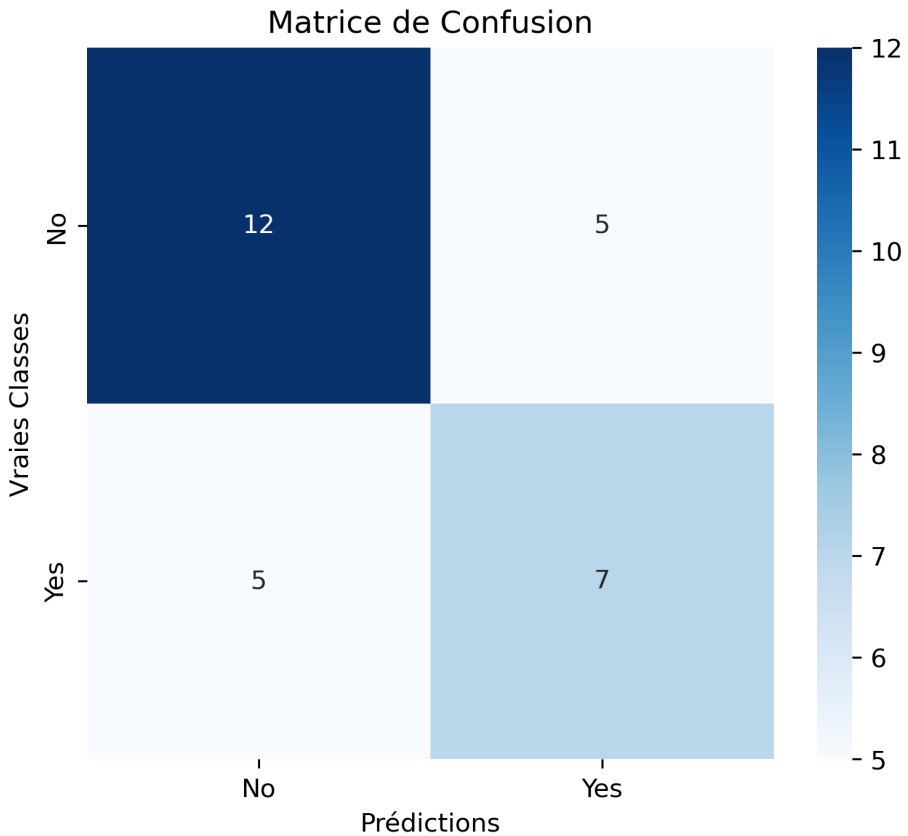


Figure: From scratch

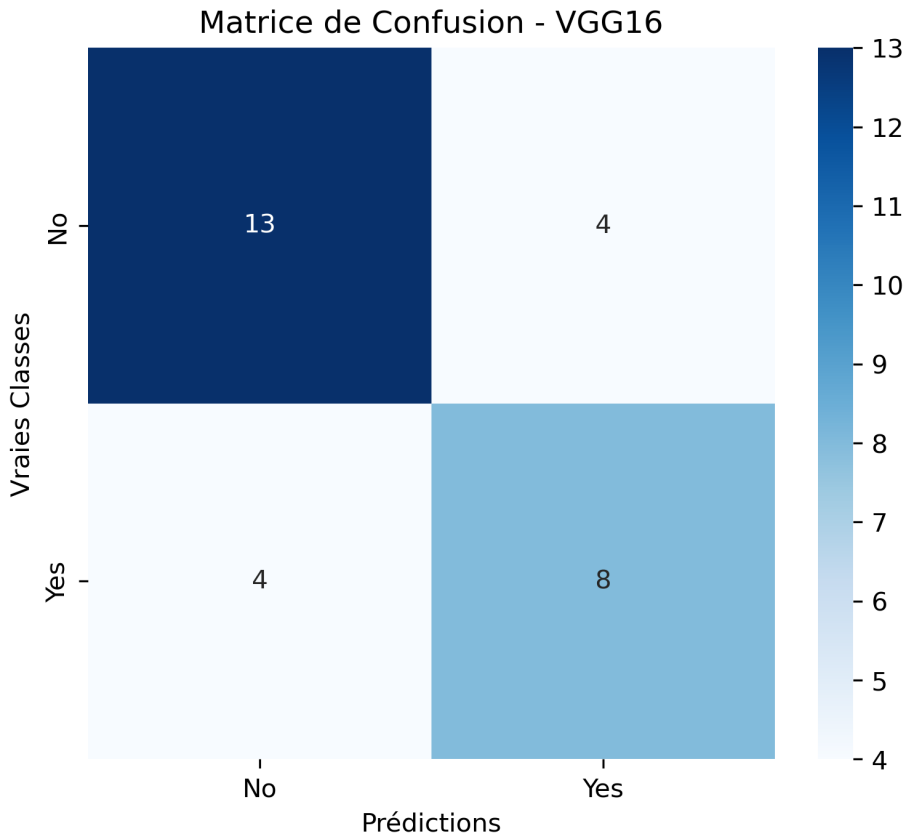


Figure: pré-entraîné sur VGG-16

Merci pour votre attention !