

# Rapport TP4

BENGHENIMA Hafsa

2CS - IASD - Groupe 02

---

## 1 Introduction

Dans un contexte où les avancées technologiques redéfinissent les pratiques médicales, l'intégration de l'intelligence artificielle dans l'analyse d'images médicales offre des solutions innovantes pour le diagnostic précoce des maladies. Ce rapport se concentre sur l'application du transfert d'apprentissage à la détection de la COVID-19 à partir d'images de radiographie thoracique. En exploitant les connaissances préalablement acquises par des modèles pré-entraînés, cette approche promet d'améliorer l'efficacité et la précision des diagnostics, ouvrant ainsi la voie à des interventions médicales plus rapides et plus précises.

## 2 Étapes du prétraitement des données

1. Chargement du jeu de données des radiographies thoraciques de COVID-19 à partir de Kaggle.
2. Normalisation des données en divisant par 255.
3. Division du jeu de données en ensembles d'entraînement (70%) et de validation (30%) en utilisant la stratégie de retenue stratifiée.

## 2 Architecture des réseaux neuronaux pré-entraînés

- **MobileNet** : MobileNet est un type de réseau de neurones conçu pour reconnaître des objets dans les images.

Pour notre étude, nous avons choisi d'utiliser **MobileNetV2** :

- **Architecture Depthwise Separable Convolution** : MobileNetV2 utilise une technique appelée "convolution Depthwise Separable" pour traiter les images plus rapidement et avec moins de mémoire.
- **Utilisation de convolutions 1×1** : Cette technique divise le travail en deux parties : regarder chaque partie de l'image séparément (convolution "depthwise"), puis mélanger toutes ces parties ensemble pour décider ce que l'image représente (convolution "pointwise").
- **Paramètres réduits**: Grâce à cette méthode, MobileNetV2 peut reconnaître les images rapidement, même sur des appareils avec moins de puissance.

### 3 Première modification

#### 3.1 Chargement des poids pré-entraînés depuis ImageNet

Nous avons chargé les poids pré-entraînés depuis ImageNet et construit un nouveau modèle en ajoutant des couches personnalisées pour s'adapter à notre tâche de classification spécifique, en utilisant la bibliothèque keras.

block_16_project (Conv2D)	(None, 7, 7, 160)	76,800	block_16_depthwise_re...
block_16_project_BN (BatchNormalization)	(None, 7, 7, 160)	640	block_16_project[0][0]
Conv_1 (Conv2D)	(None, 7, 7, 1280)	204,800	block_16_project_BN[0...
Conv_1_bn (BatchNormalization)	(None, 7, 7, 1280)	5,120	Conv_1[0][0]
out_relu (ReLU)	(None, 7, 7, 1280)	0	Conv_1_bn[0][0]

Total params: 706,224 (2.69 MB)

Trainable params: 687,680 (2.62 MB)

Non-trainable params: 18,544 (72.44 KB)

#### 3.2 Remplacement des couches entièrement connectées

Les dernières couches entièrement connectées du modèle pré-entraîné ont été supprimées et remplacées par trois nouvelles couches entièrement connectées, comportant respectivement **1024**, **512**, et **4** neurones, ce dernier correspondant au **nombre de classes** de notre tâche de classification.

out_relu (ReLU)	(None, 7, 7, 1280)	0	Conv_1_bn[0][...	-
global_average_p... (GlobalAveragePo...	(None, 1280)	0	out_relu[0][0]	-
dense (Dense)	(None, 1024)	1,311,744	global_averag...	Y
dense_1 (Dense)	(None, 512)	524,800	dense[0][0]	Y
dense_2 (Dense)	(None, 4)	2,052	dense_1[0][0]	Y

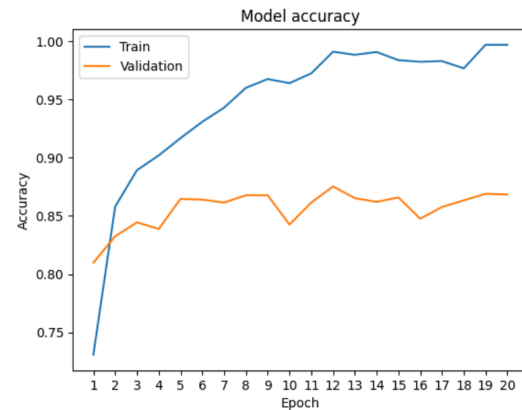
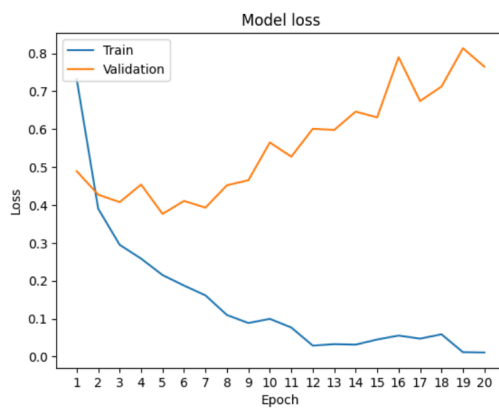
Total params: 2,544,820 (9.71 MB)

Trainable params: 1,838,596 (7.01 MB)

Non-trainable params: 706,224 (2.69 MB)

•

**Explication** : En ajustant uniquement les couches entièrement connectées finales, nous permettons au modèle de se spécialiser dans la classification des caractéristiques extraites par les couches convolutionnelles pré-entraînées.



## 4 Deuxieme modification

### 4.1 One convolutional layers + the last three fully connected

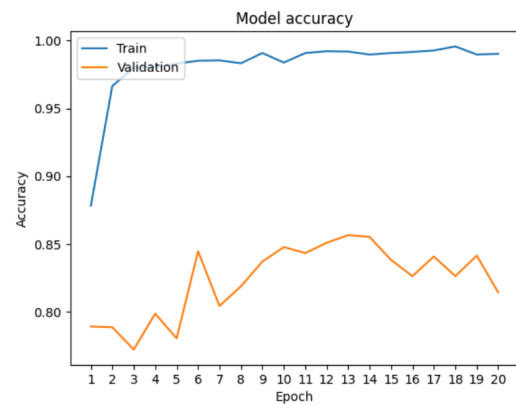
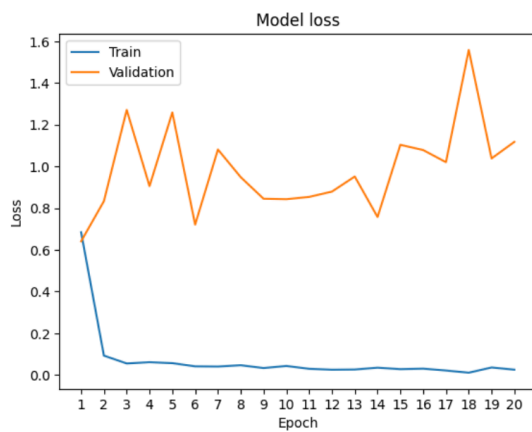
Conv_1 (Conv2D)	(None, 7, 7, 1280)	204,800	block_16_proj...	Y
Conv_1_bn (BatchNormalizat...	(None, 7, 7, 1280)	5,120	Conv_1[0][0]	Y
out_relu (ReLU)	(None, 7, 7, 1280)	0	Conv_1_bn[0][...	-
global_average_p... (GlobalAveragePo...	(None, 1280)	0	out_relu[0][0]	-
dense (Dense)	(None, 1024)	1,311,744	global_averag...	Y
dense_1 (Dense)	(None, 512)	524,800	dense[0][0]	Y
dense_2 (Dense)	(None, 4)	2,052	dense_1[0][0]	Y

Total params: 2,544,820 (9.71 MB)

Trainable params: 2,045,956 (7.80 MB)

Non-trainable params: 498,864 (1.90 MB)

Second model



- **Explication** : En autorisant l'ajustement des couches convolutionnelles, le modèle peut capturer des caractéristiques spécifiques à la tâche qui n'étaient pas bien représentées dans les caractéristiques pré-entraînées.

## 5 Troisième modification

### 5.1 Two convolutional layers + the last three fully connected

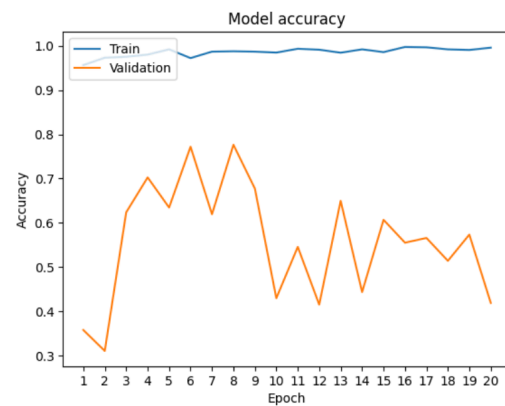
block_16_project (Conv2D)	(None, 7, 7, 160)	76,800	block_16_dept...	Y
block_16_project... (BatchNormalizat...	(None, 7, 7, 160)	640	block_16_proj...	Y
Conv_1 (Conv2D)	(None, 7, 7, 1280)	204,800	block_16_proj...	Y
Conv_1_bn (BatchNormalizat...	(None, 7, 7, 1280)	5,120	Conv_1[0][0]	Y
out_relu (ReLU)	(None, 7, 7, 1280)	0	Conv_1_bn[0][...	-
global_average_p... (GlobalAveragePo...	(None, 1280)	0	out_relu[0][0]	-
dense (Dense)	(None, 1024)	1,311,744	global_averag...	Y
dense_1 (Dense)	(None, 512)	524,800	dense[0][0]	Y
dense_2 (Dense)	(None, 4)	2,052	dense_1[0][0]	Y

Total params: 2,544,820 (9.71 MB)

Trainable params: 2,123,076 (8.10 MB)

Non-trainable params: 421,744 (1.61 MB)

Third model



**Explication :** L'introduction de couches convolutionnelles entraînables supplémentaires peut conduire à un surajustement et à une diminution de la capacité du modèle à généraliser à de nouvelles données, ce qui se traduit par une baisse des performances.

## 6 Résultats

Model	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss	Training Time (s)
Only the last 3 FC	0.997029	0.868388	0.0108198	0.764598	98.5047
The last Conv	0.990278	0.814232	0.0256406	1.11791	93.8622
The last 2 Conv	0.995679	0.419395	0.0117113	13.6389	95.6827

- **Three Connected Layers**

Class	Precision	Recall	F1-Score	Accuracy
COVID	0.87	0.82	0.84	0.88
Lung_Opacity	0.84	0.86	0.85	
Normal	0.89	0.90	0.89	
Viral Pneumonia	0.97	0.94	0.95	

- **One Convolutional Layer + Three Connected Layers**

Class	Precision	Recall	F1-Score	Accuracy
COVID	0.93	0.73	0.82	0.84
Lung_Opacity	0.73	0.88	0.80	
Normal	0.90	0.85	0.87	
Viral Pneumonia	0.90	0.91	0.91	

- **Two Convolutional Layers + Three Connected Layers**

Class	Precision	Recall	F1-Score	Accuracy
COVID	0.81	0.62	0.70	0.60
Lung_Opacity	0.43	0.95	0.59	
Normal	0.94	0.43	0.59	
Viral Pneumonia	0.97	0.33	0.49	

## 7 Conclusions Générales

- Le transfert learning consiste à utiliser des poids pré-entraînés sur de grandes bases de données. Il permet au modèle d'initialiser ses poids avec des connaissances préalables qui peuvent être adaptées à notre tâche spécifique de classification des rayons X.

- En ajustant différentes parties du modèle, nous influençons sa capacité à reconnaître les caractéristiques importantes pour distinguer les patients atteints de la COVID-19 de ceux en bonne santé.
- Les couches convolutionnelles peuvent être ajustées pour capturer des caractéristiques spécifiques aux rayons X des patients atteints de COVID-19.
- Lorsque nous ajustons le modèle, le nombre total de paramètres peut changer. Par exemple, l'ajout de couches entraînaibles peut augmenter ce nombre, ce qui peut être utile pour capturer des caractéristiques plus complexes.
- Trop d'ajustements peuvent entraîner un surajustement, rendant le modèle moins efficace pour de nouvelles données.
- Sur la base de nos résultats, le modèle avec les trois dernières couches entièrement connectées entraînaibles semble être le meilleur choix. Ce modèle offre un bon équilibre entre spécialisation et capacité de généralisation, ce qui se traduit par de bonnes performances sur l'ensemble de validation.

## Partie 02 : Segmentation

### 1. Architecture du Modèle

- **Base** : U-Net
- **Taille d'entrée** : (128, 128, 3)
- **Composants clés** :
  - Encodeur + Décodeur avec connexions de saut (skip connections)
  - Convolutions 3×3, max pooling 2×2
  - Activations ReLU, sortie avec sigmoid
  - Précision mixte activée pour accélérer l'entraînement

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 128, 128, 3)	0	-
conv2d (Conv2D)	(None, 128, 128, 32)	896	input_layer[0][0]
conv2d_1 (Conv2D)	(None, 128, 128, 32)	9,248	conv2d[0][0]
max_pooling2d (MaxPooling2D)	(None, 64, 64, 32)	0	conv2d_1[0][0]
conv2d_2 (Conv2D)	(None, 64, 64, 64)	18,496	max_pooling2d[0][0]
conv2d_3 (Conv2D)	(None, 64, 64, 64)	36,928	conv2d_2[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 64)	0	conv2d_3[0][0]
conv2d_4 (Conv2D)	(None, 32, 32, 128)	73,856	max_pooling2d_1[0][0]
conv2d_5 (Conv2D)	(None, 32, 32, 128)	147,584	conv2d_4[0][0]
conv2d_transpose (Conv2DTranspose)	(None, 64, 64, 64)	32,832	conv2d_5[0][0]
concatenate (Concatenate)	(None, 64, 64, 128)	0	conv2d_transpose[0][0]... conv2d_3[0][0]
conv2d_6 (Conv2D)	(None, 64, 64, 64)	73,792	concatenate[0][0]
conv2d_7 (Conv2D)	(None, 64, 64, 64)	36,928	conv2d_6[0][0]
conv2d_transpose_1 (Conv2DTranspose)	(None, 128, 128, 32)	8,224	conv2d_7[0][0]
concatenate_1 (Concatenate)	(None, 128, 128, 64)	0	conv2d_transpose_1[0]... conv2d_1[0][0]
conv2d_8 (Conv2D)	(None, 128, 128, 32)	18,464	concatenate_1[0][0]
conv2d_9 (Conv2D)	(None, 128, 128, 32)	9,248	conv2d_8[0][0]
conv2d_10 (Conv2D)	(None, 128, 128, 1)	33	conv2d_9[0][0]

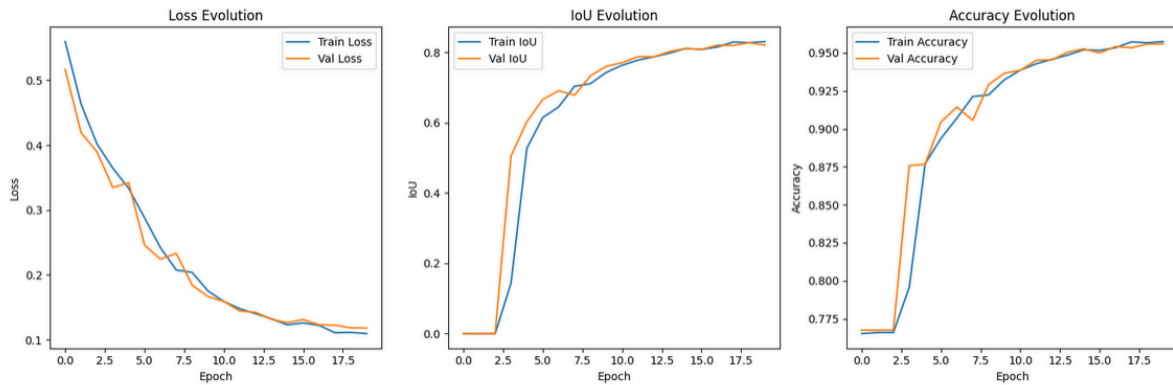
## 2. Entraînement

- **Fonction de coût** : Binary Crossentropy
- **Optimiseur** : Adam (avec `LossScaleOptimizer`)
- **Métriques suivies** : IoU binaire, exactitude, rappel, précision
- **Nombre d'époques** : 20
- **Taille du batch** : 16
- **Split des données** : 70% entraînement / 30% validation, stratifié par classe

## 3. Résultats d'Entraînement

### Courbes d'entraînement





### Final Validation Metrics:

F1-Score: 0.9013

Precision: 0.9392

Recall: 0.8664

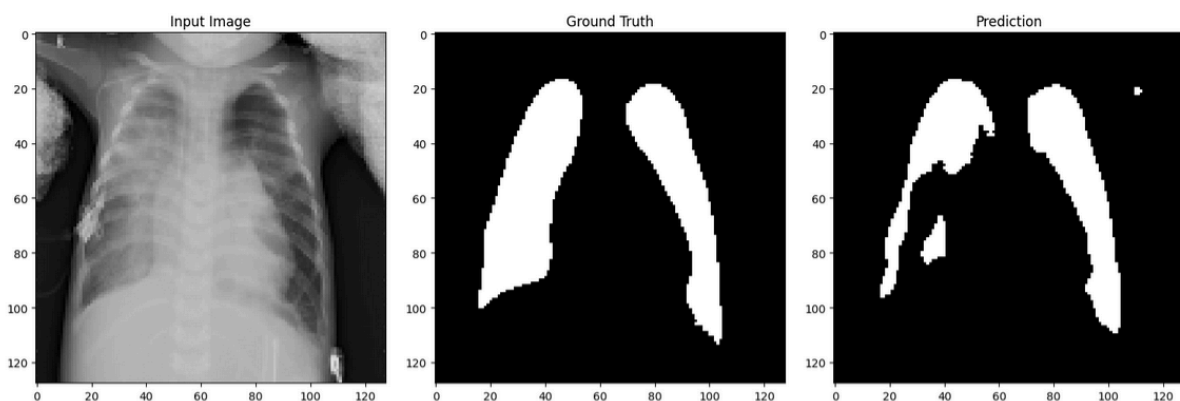
IoU: 0.8204

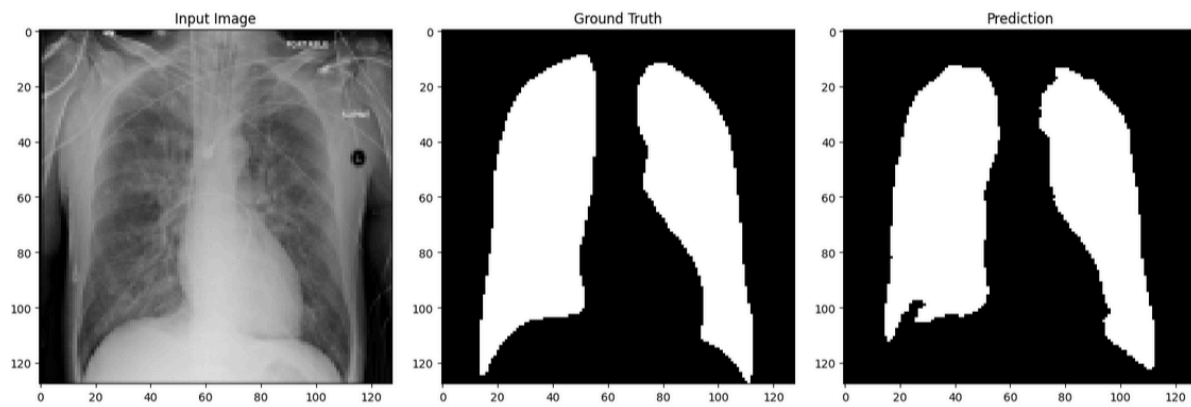
### Observations :

- Les courbes montrent une amélioration constante des performances.
- Aucune divergence ou surapprentissage manifeste. Le modèle apprend bien sans "mémoriser" le jeu d'entraînement.
- L'IoU élevé ( $> 0.80$ ) indique une bonne capacité de **généralisation** du modèle.
- Le modèle **U-Net** a bien appris à segmenter les images de radiographies thoraciques, avec des **métriques de segmentation solides**.

## 4. Évaluation Qualitative

### Prédictions Visuelles





### Observations :

- Le masque prédit est **globalement très proche** de la vérité terrain.
- Les **contours sont légèrement flous** ou imprécis sur certaines zones, notamment dans les régions inférieures du poumon.
- Le modèle **réussit à détecter correctement les deux lobes**, sans fusion ni déformation majeures.