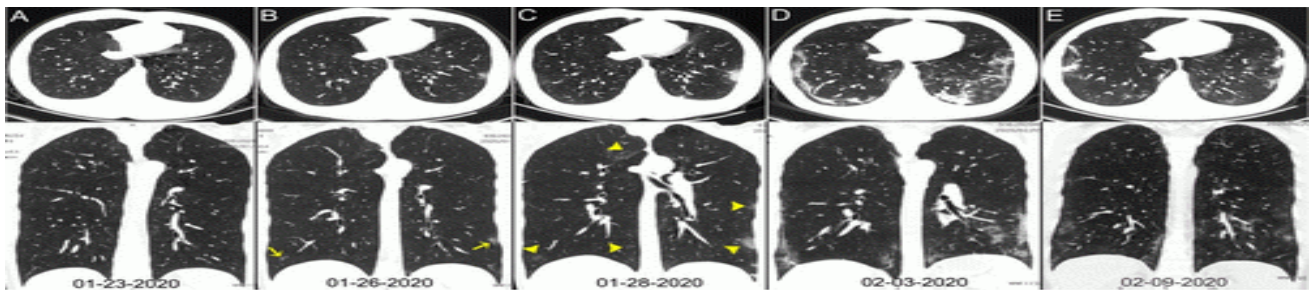
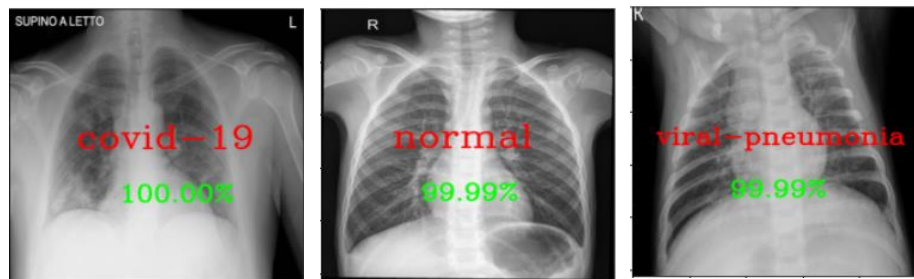


## Intelligence Artificielle (I-ILIA-026) -- Travaux pratiques

### TP 05 : CNN et Transfer Learning pour la classification d'images Covid-19

**A remettre**

L'objectif de ce TP est de développer un réseau de neurones profond pour classifier des images médicales (radiographies) et détecter la présence ou pas de pathologie Covid-19. Plus concrètement, le modèle développé devra identifier la classe d'une image radiographique (X-ray) parmi trois possibilités : covid-19, normal, viral-pneumonia (Fig. 1). Pour réaliser ce TP, nous vous proposons d'exploiter les réseaux de neurones convolutionnels (CNN) ainsi que le principe du Transfer Learning.



**Figure** : exemples d'images X-ray et CT-scans : covid-19, normal et viral-pneumonia

Le TP peut être réalisé en six parties :

#### I. Partie 1 : Environnement et téléchargement de données

1. **Vérification du matériel** : vérifier que le GPU est bien sélectionné ;
2. **Importation des libraires** : importer les libraires nécessaires ;
3. **Télécharger la base d'images radiographiques** : représentée par trois classes (covid-19, normal et viral-pneumonia). Cette base de données est composée d'un ensemble d'entraînement (train) et de test. Il faudra donc diviser les données d'entraînement pour extraire les données de validation (point. 8 et .9)

## II. Partie 2 : Prétraitement des données et paramètres d'entraînement

4. Créer le fichier de labels : avec 3 classes
5. Définir les paramètres d'entraînement
6. Analyser/exécuter la cellule permettant d'associer les labels aux images correspondantes
7. Charger et prétraiter les données d'entraînement 80% (adaptation du format de labels « One Hot »)
8. Charger et prétraiter les données de validation 20% (adaptation du format de labels « One Hot »)

## III. Partie 3 : Entraînement du modèle convolutionnel CNN « Transfer Learning »:

9. Télécharge le modèle « MobileNet » pré-entraîné avec « ImageNet » avant de l'adapter pour notre problème de classification à 3 classes (approche de Transfer Learning)
10. Compiler et entraîner le modèle et analyser les résultats
11. Sauvegarder le modèle et afficher les courbes d'apprentissage
12. Tester et visualiser le résultat du modèle avec une image de test
13. Charger et prétraiter la base de données de test (300 images)
14. Evaluer le modèle avec cette base de données de test
15. Identifier des erreurs de test et générer du fichier .csv à soumettre pour le challenge Kaggle
16. Adapter les paramètres (epochs, lr, batch size, etc.) en fonction des résultats et entraîner de nouveau
17. Tester d'autres architectures (Xception, ResNet50, DenseNet121, etc.)
18. Analyser et interpréter les résultats
19. Calculer la matrice de confusion pour votre meilleur modèle.

## IV. Partie 4 : Visualisation et explicabilité

20. Choisir le (ou les) meilleur(s) modèle(s) en termes de résultat et analyser/visualiser les couches convolutionnelles (récupérés avec « model.layers » et visualisés avec plot)
21. Appliquer les techniques d'explicabilité (GradCAM, LRP, GuidedBackPropagation, etc.) afin d'identifier les pixels ou régions responsables aux décisions de vos modèles

**Note :** pour appliquer ces techniques, il faudra installer le module d'explicabilité « tf-explain »

## V. Partie 5 : Analyse élargie (facultatif)

22. Développer votre propre modèle « *From Scratch* » et comparer avec les modèles développés ci-dessus
23. Tester les modèles avec d'autres bases de données publiques d'images radiographiques (voir Moodle)
24. Proposer un modèle pour la classification d'images de scan (CT-Scan) pour la même problématique.

**PS :** vous avez le libre choix de bases de données (X-ray ou CT-Scans)

➤ **A remettre :** pour le 28/11/2020 sur Moodle

Il faudra remettre deux éléments :

1. Tableau comparatif complété

Caractéristiques	Modèle 1	Modèle 2	Modèle 3	....	Modèle N
<b>Entrainement</b>					
Taille du modèle en MB					
Training_Loss					
Training_Accuracy					
Validation_Loss					
Validation_Accuracy					
Temps (optionnel)					
<b>Test</b>					
Test_Loss					
Test_Accuracy					
Nombre d'erreurs					
Classement Kaggle (1 colonne)					

**Note :** En cas d'utilisation de plusieurs datasets, il faudra fournir **un tableau par dataset**.

2. Meilleur modèle (.h5) avec indication de la taille d'entrée du réseau.
3. Le Notebook utilisé pour l'entraînement et test des modèles.
4. Votre conclusion sur l'étude en termes de précision, temps de calcul et explicabilité de modèle  
(1 page au maximum)

- **Lien du challenge Kaggle :** <https://www.kaggle.com/c/covid-19-challenge-umons>
- **Note 1 :** un code de démarrage est partagé via Moodle « **Input\_TP5\_IA\_2021.ipynb** »
- **Note 2 :** les travaux sont à remettre par groupe

**ANNEXES :**

- Autres bases de données :**

a. Small Dataset (3 classes): [http://172.104.134.59/covid\\_dataset\\_3classes.tar](http://172.104.134.59/covid_dataset_3classes.tar)

b. Dataset (2 classes) : [http://172.104.134.59/covid\\_dataset\\_2classes.tar](http://172.104.134.59/covid_dataset_2classes.tar)

c. Dataset (4 classes): [http://172.104.134.59/covid\\_dataset\\_4classes.tar](http://172.104.134.59/covid_dataset_4classes.tar)

d. Big Dataset (4 classes): <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>

e. CT-scan Dataset: <https://github.com/UCSD-AI4H/COVID-CT>

- Exemple de modèle From Scratch :**

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 224, 224, 64)	1792
conv2d_1 (Conv2D)	(None, 224, 224, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 112, 112, 64)	0
conv2d_2 (Conv2D)	(None, 112, 112, 128)	73856
conv2d_3 (Conv2D)	(None, 112, 112, 128)	147584
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 128)	0
flatten (Flatten)	(None, 401408)	0
dense (Dense)	(None, 512)	205521408
activation (Activation)	(None, 512)	0
dense_1 (Dense)	(None, 3)	1539
activation_1 (Activation)	(None, 3)	0

- Exemple de modèle From Scratch :**

