Wheat Leaf diseases classification project report

- **IA model :** VGG19 pretrained model
- **Dataset classes :** 4 classes [‘Crown & Root Rot’, Healthy , Leaf_rsut, Wheat Loose Smut]

```
number of categories:  4
```

	categorie	numbre of files
0	Crown and Root Rot	696
1	Healthy Wheat	1030
2	Leaf Rust	849
3	Wheat Loose Smut	939

- **Train model:**

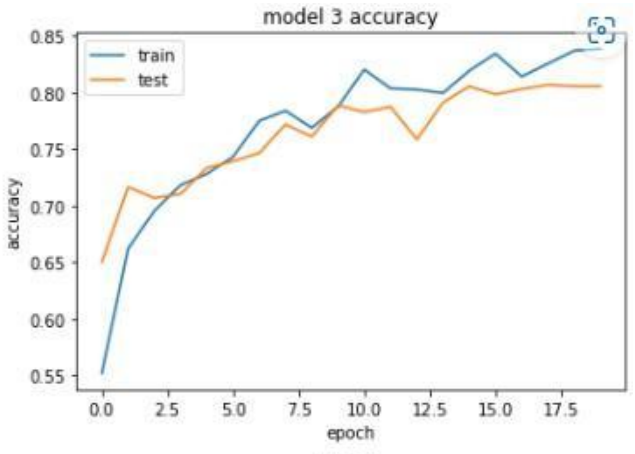
```
[21] # initialized with actual "learned" values versus pure random
     H = moodel.fit(
         train_generator.flow(x_train, y_train, batch_size=32),
         steps_per_epoch=len(x_train) // 32,
         validation_data=val_generator.flow(x_val,y_val),
         validation_steps=len(x_val) // 32,
         epochs=20)
```

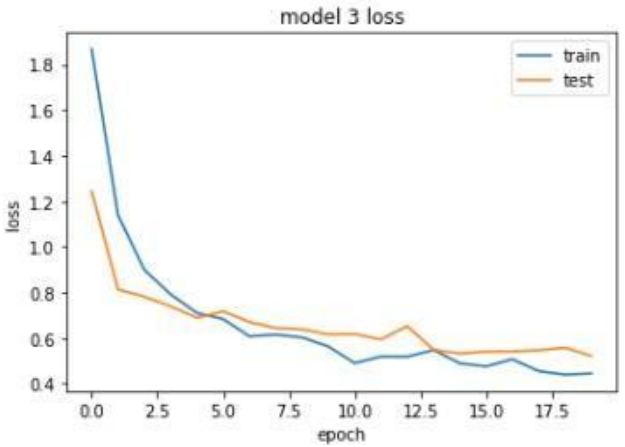
```
Epoch 1/20
61/61 [==============================] - 75s 965ms/step - loss: 1.8676 - accuracy: 0.5523 - val_loss: 1.2437 - val_accuracy: 0.6502
Epoch 2/20
61/61 [==============================] - 58s 946ms/step - loss: 1.1393 - accuracy: 0.6620 - val_loss: 0.8134 - val_accuracy: 0.7163
Epoch 3/20
61/61 [==============================] - 59s 969ms/step - loss: 0.8997 - accuracy: 0.6953 - val_loss: 0.7817 - val_accuracy: 0.7067
Epoch 4/20
61/61 [==============================] - 58s 954ms/step - loss: 0.7921 - accuracy: 0.7181 - val_loss: 0.7396 - val_accuracy: 0.7103
Epoch 5/20
61/61 [==============================] - 58s 950ms/step - loss: 0.7101 - accuracy: 0.7280 - val_loss: 0.6890 - val_accuracy: 0.7332
Epoch 6/20
61/61 [==============================] - 58s 956ms/step - loss: 0.6836 - accuracy: 0.7431 - val_loss: 0.7171 - val_accuracy: 0.7392
Epoch 7/20
61/61 [==============================] - 58s 954ms/step - loss: 0.6079 - accuracy: 0.7748 - val_loss: 0.6705 - val_accuracy: 0.7464
Epoch 8/20
61/61 [==============================] - 58s 948ms/step - loss: 0.6156 - accuracy: 0.7837 - val_loss: 0.6442 - val_accuracy: 0.7716
Epoch 9/20
61/61 [==============================] - 58s 952ms/step - loss: 0.6029 - accuracy: 0.7686 - val_loss: 0.6383 - val_accuracy: 0.7608
Epoch 10/20
61/61 [==============================] - 58s 951ms/step - loss: 0.5634 - accuracy: 0.7873 - val_loss: 0.6164 - val_accuracy: 0.7885
Epoch 11/20
61/61 [==============================] - 58s 953ms/step - loss: 0.4902 - accuracy: 0.8201 - val_loss: 0.6179 - val_accuracy: 0.7825
Epoch 12/20
61/61 [==============================] - 59s 975ms/step - loss: 0.5190 - accuracy: 0.8034 - val_loss: 0.5943 - val_accuracy: 0.7873
Epoch 13/20
61/61 [==============================] - 60s 977ms/step - loss: 0.5187 - accuracy: 0.8024 - val_loss: 0.6526 - val_accuracy: 0.7584
Epoch 14/20
61/61 [==============================] - 59s 965ms/step - loss: 0.5477 - accuracy: 0.7993 - val_loss: 0.5487 - val_accuracy: 0.7909
Epoch 15/20
61/61 [==============================] - 59s 964ms/step - loss: 0.4894 - accuracy: 0.8190 - val_loss: 0.5320 - val_accuracy: 0.8053
Epoch 16/20
61/61 [==============================] - 58s 957ms/step - loss: 0.4760 - accuracy: 0.8341 - val_loss: 0.5410 - val_accuracy: 0.7981
Epoch 17/20
61/61 [==============================] - 59s 967ms/step - loss: 0.5070 - accuracy: 0.8138 - val_loss: 0.5417 - val_accuracy: 0.8029
Epoch 18/20
61/61 [==============================] - 59s 974ms/step - loss: 0.4551 - accuracy: 0.8253 - val_loss: 0.5465 - val_accuracy: 0.8065
Epoch 19/20
61/61 [==============================] - 68s 1s/step - loss: 0.4400 - accuracy: 0.8367 - val_loss: 0.5582 - val_accuracy: 0.8053
```

émoire RAM disponible. Afficher les journaux d'exécution ×

```
71ms/step - loss: 0.4449 - accuracy: 0.8383 - val_loss: 0.5219 - val_accuracy: 0.8053
```

- **Curves :**





- **Confusion matrix :**

```
[27] from sklearn.metrics import classification_report
     print(classification_report(y_true, y_pred))
```

```
                  precision    recall  f1-score   support

               0       0.84      0.76      0.80       147
               1       0.81      0.79      0.80       199
               2       0.84      0.83      0.84       178
               3       0.79      0.89      0.83       175

        accuracy                           0.82       699
       macro avg       0.82      0.82      0.82       699
    weighted avg       0.82      0.82      0.82       699
```

```
[28] # get confusion matrix
     confusion_mat = confusion_matrix(y_true, y_pred)
     # plot confusion_mat
     plt_confusion_mat(confusion_mat, classes=categories, fig_size=(20, 7))
```