UNINA - MACHINE LEARNING 21/22 - FINAL CONTEST

Fibrosarcoma cells classification

GAROFALO LUIGI
M63001182

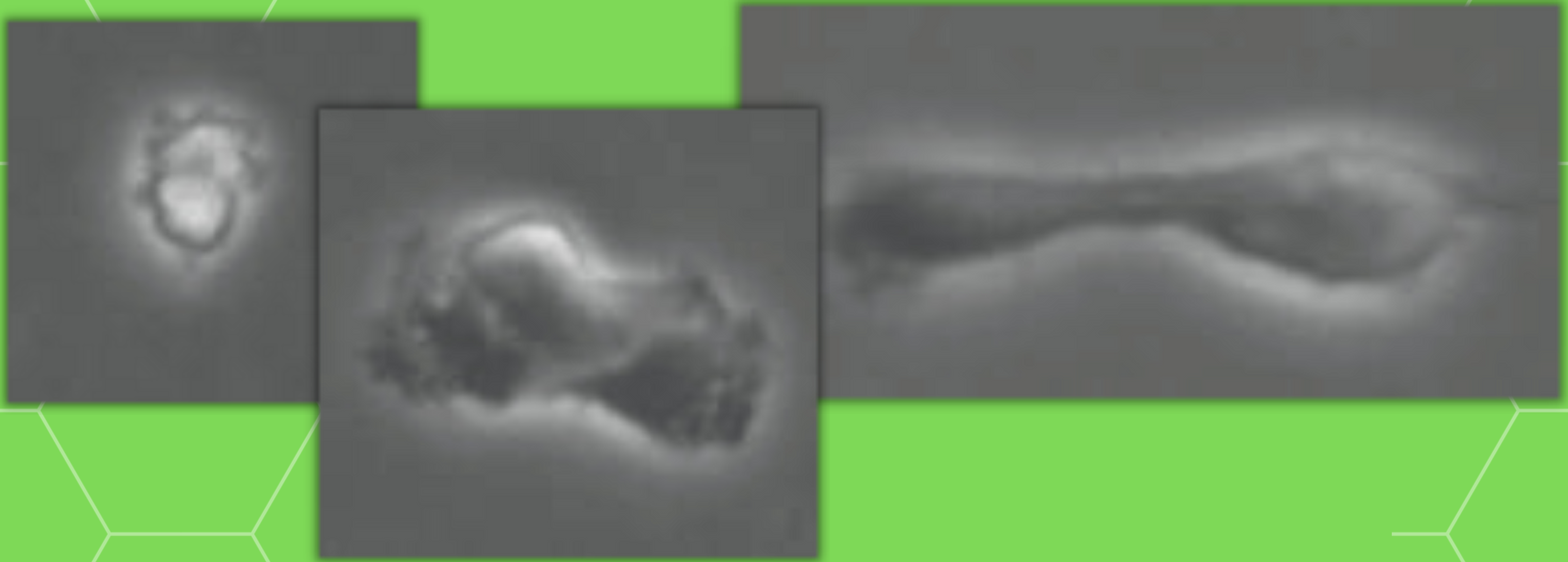
BRESCIA MATTIA
M63001103

INTRODUCTION

FIBROSARCOMA IS A MALIGNANT TUMOUR AFFECTING THE FIBROUS CONNECTIVE TISSUE.

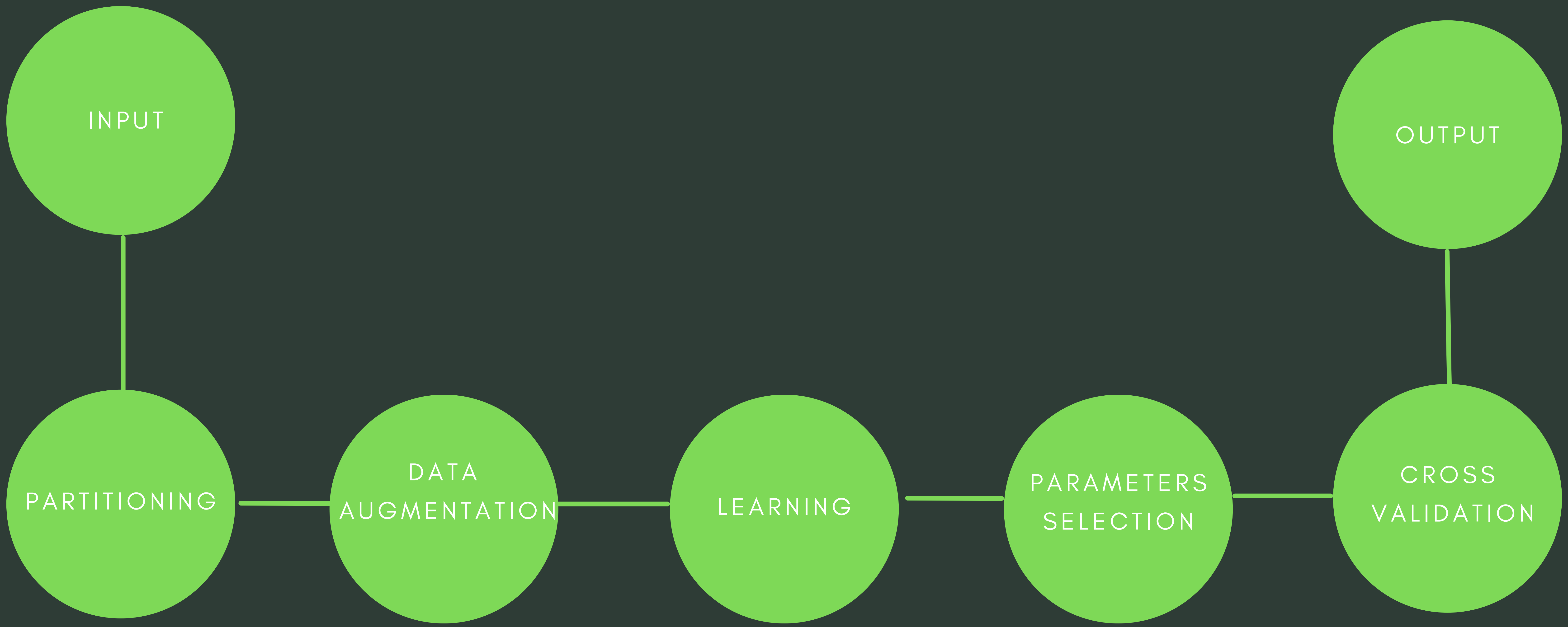
The aim of the final Machine Learning contest for the academic year 2021/2022 is to recognise four different cell types (multi-class classification) according to their morphology.

DATASET



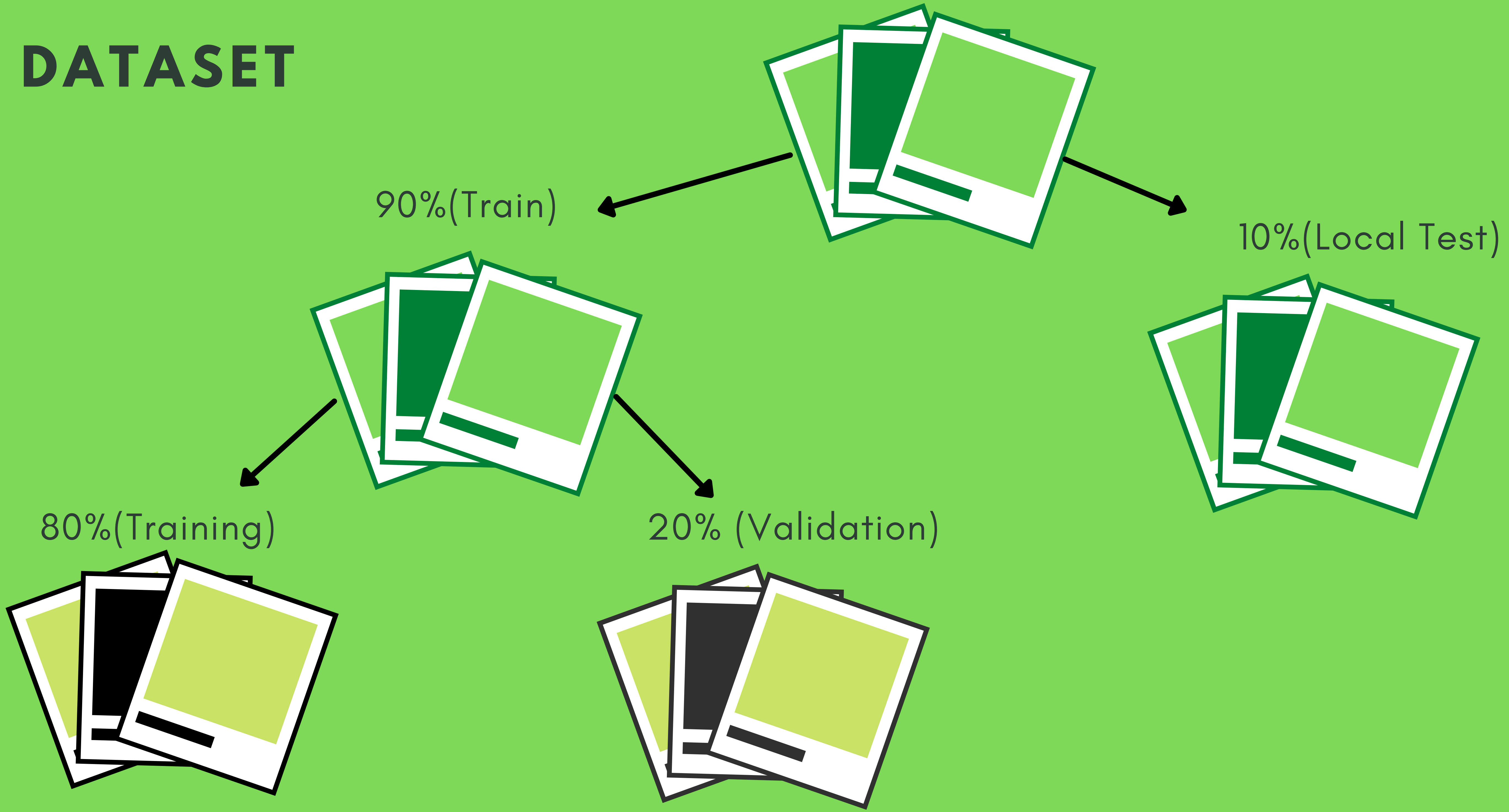
- 1374 TRAINING IMAGES AND LABELS
- 344 TEST IMAGES

PROCESS


INPUT
PARTITIONING
DATA AUGMENTATION
LEARNING
PARAMETERS SELECTION
CROSS VALIDATION
OUTPUT

PARTITIONING OF

THE DATASET



90%(Train)

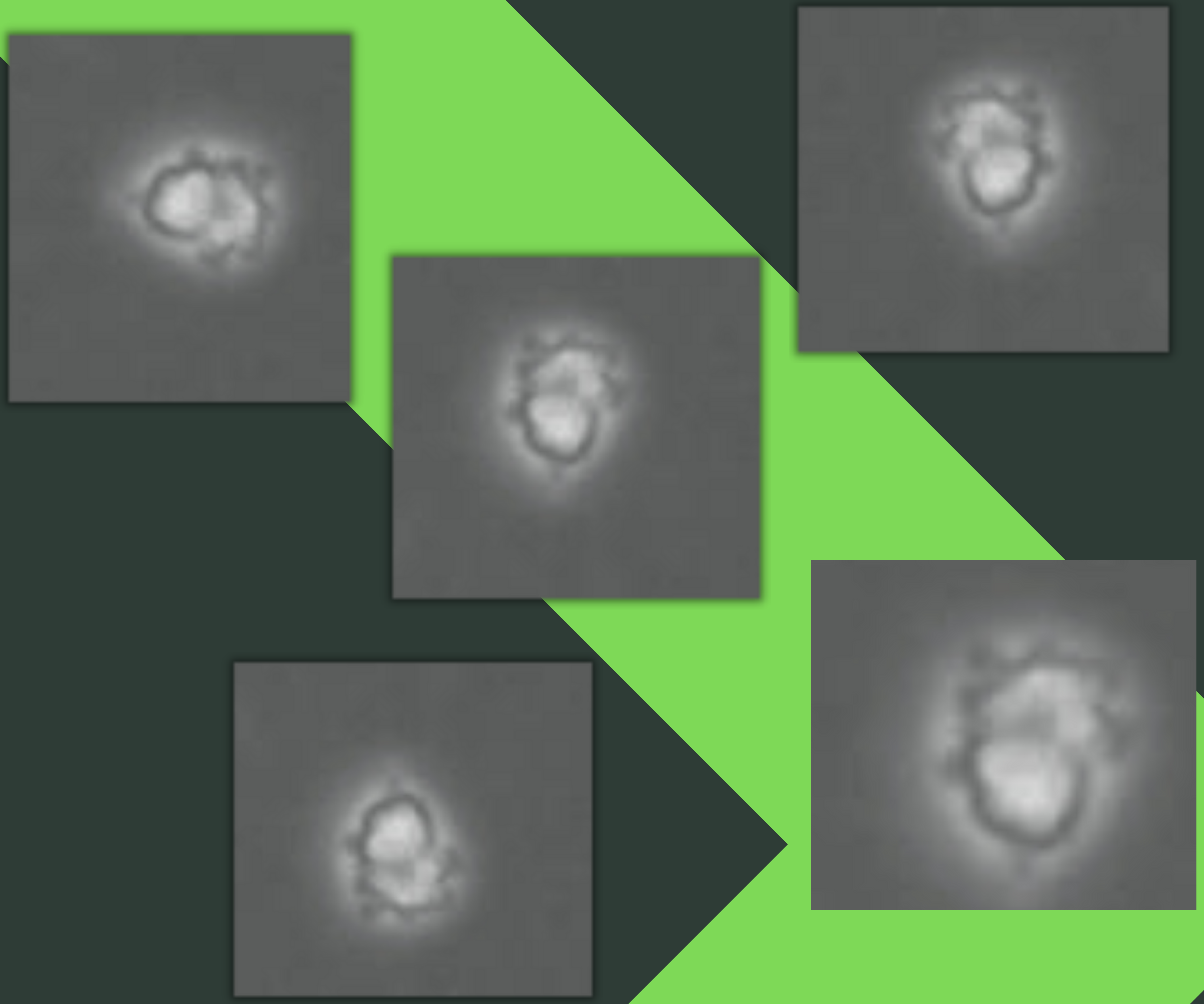
10%(Local Test)

80%(Training)

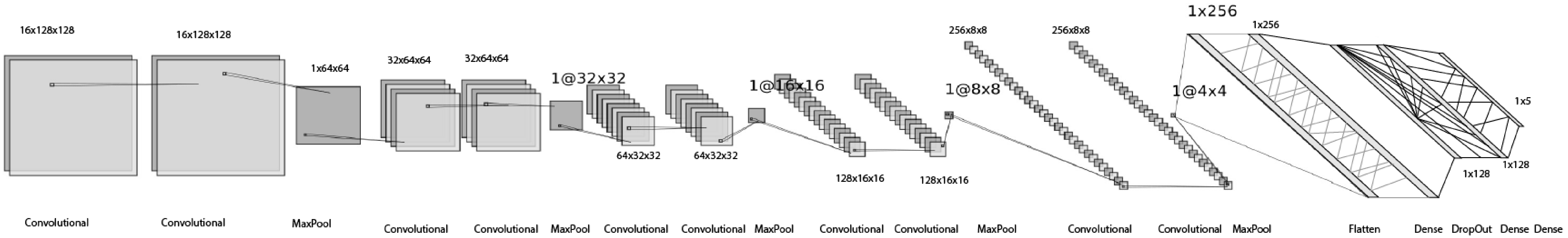
20% (Validation)

DATA AUGMENTATION

- VERTICAL/HORIZONTAL FLIP
- ROTATION (90°)
- ZOOM [0.7,1.3]
- FEATUREWISE NORMALIZATION(CENTER,STD)

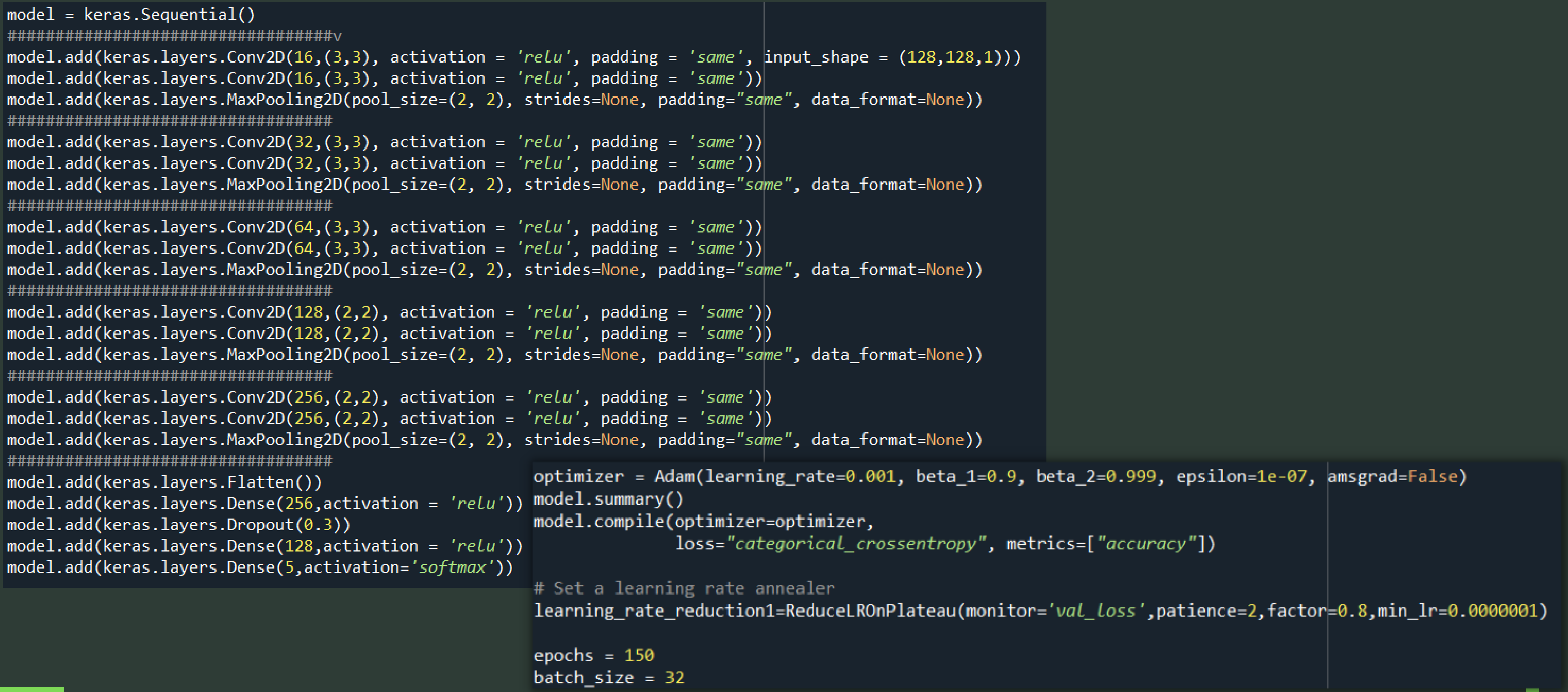


MODEL



ADAM OPTIMIZER

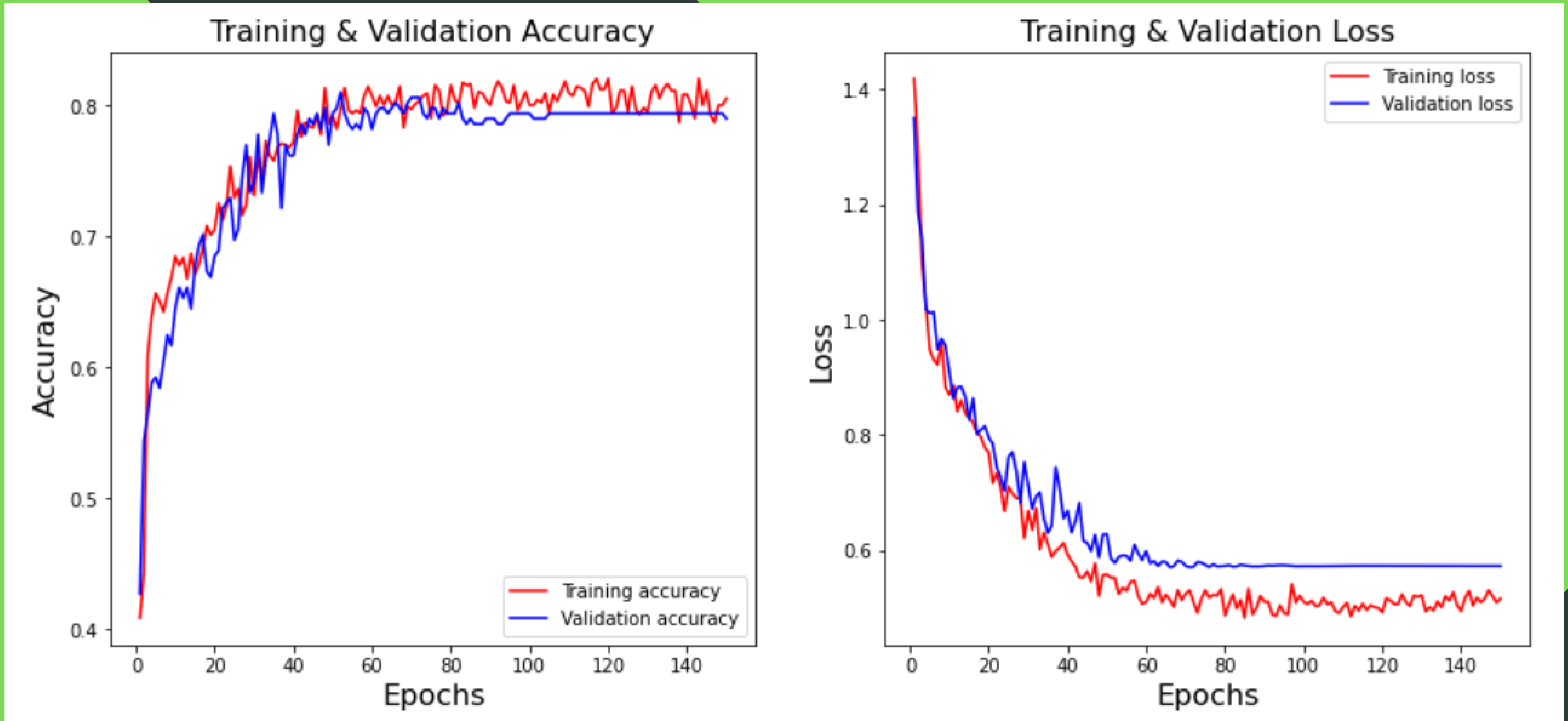
MODEL



```
model = keras.Sequential()
##################################v
model.add(keras.layers.Conv2D(16,(3,3), activation = 'relu', padding = 'same', input_shape = (128,128,1)))
model.add(keras.layers.Conv2D(16,(3,3), activation = 'relu', padding = 'same'))
model.add(keras.layers.MaxPooling2D(pool_size=(2, 2), strides=None, padding="same", data_format=None))
##################################
model.add(keras.layers.Conv2D(32,(3,3), activation = 'relu', padding = 'same'))
model.add(keras.layers.Conv2D(32,(3,3), activation = 'relu', padding = 'same'))
model.add(keras.layers.MaxPooling2D(pool_size=(2, 2), strides=None, padding="same", data_format=None))
##################################
model.add(keras.layers.Conv2D(64,(3,3), activation = 'relu', padding = 'same'))
model.add(keras.layers.Conv2D(64,(3,3), activation = 'relu', padding = 'same'))
model.add(keras.layers.MaxPooling2D(pool_size=(2, 2), strides=None, padding="same", data_format=None))
##################################
model.add(keras.layers.Conv2D(128,(2,2), activation = 'relu', padding = 'same'))
model.add(keras.layers.Conv2D(128,(2,2), activation = 'relu', padding = 'same'))
model.add(keras.layers.MaxPooling2D(pool_size=(2, 2), strides=None, padding="same", data_format=None))
##################################
model.add(keras.layers.Conv2D(256,(2,2), activation = 'relu', padding = 'same'))
model.add(keras.layers.Conv2D(256,(2,2), activation = 'relu', padding = 'same'))
model.add(keras.layers.MaxPooling2D(pool_size=(2, 2), strides=None, padding="same", data_format=None))
##################################
model.add(keras.layers.Flatten())
model.add(keras.layers.Dense(256,activation = 'relu'))
model.add(keras.layers.Dropout(0.3))
model.add(keras.layers.Dense(128,activation = 'relu'))
model.add(keras.layers.Dense(5,activation='softmax'))
```

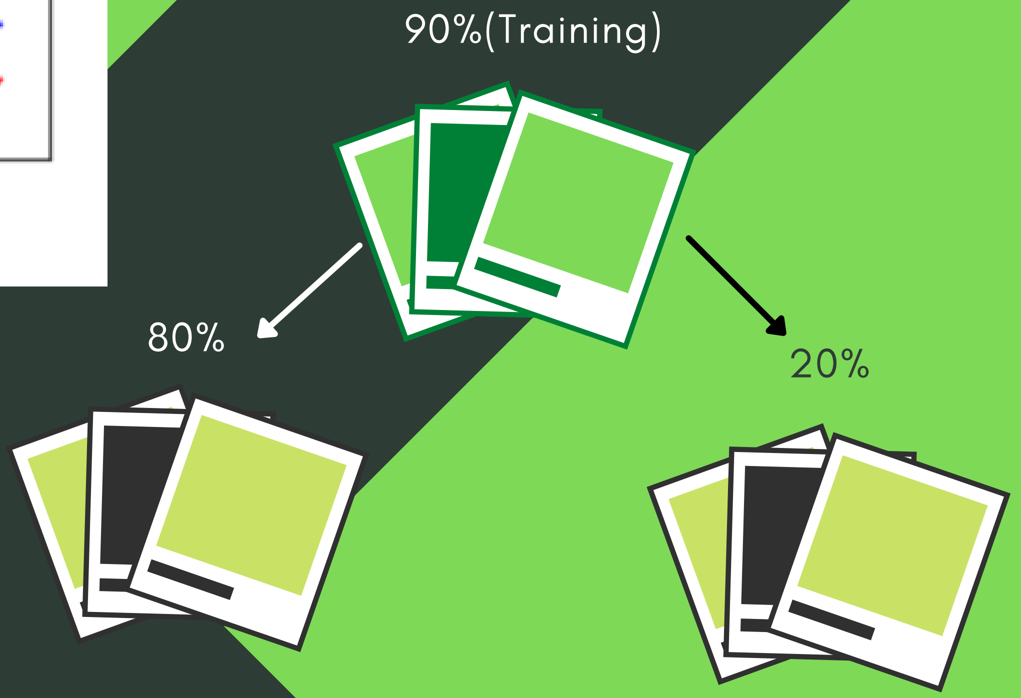
```
optimizer = Adam(learning_rate=0.001, beta_1=0.9, beta_2=0.999, epsilon=1e-07, amsgrad=False)
model.summary()
model.compile(optimizer=optimizer,
              loss="categorical_crossentropy", metrics=["accuracy"])

# Set a learning rate annealer
learning_rate_reduction1=ReduceLROnPlateau(monitor='val_loss',patience=2,factor=0.8,min_lr=0.0000001)

epochs = 150
batch_size = 32
```

RESULTS



Training & Validation Accuracy

Training & Validation Loss

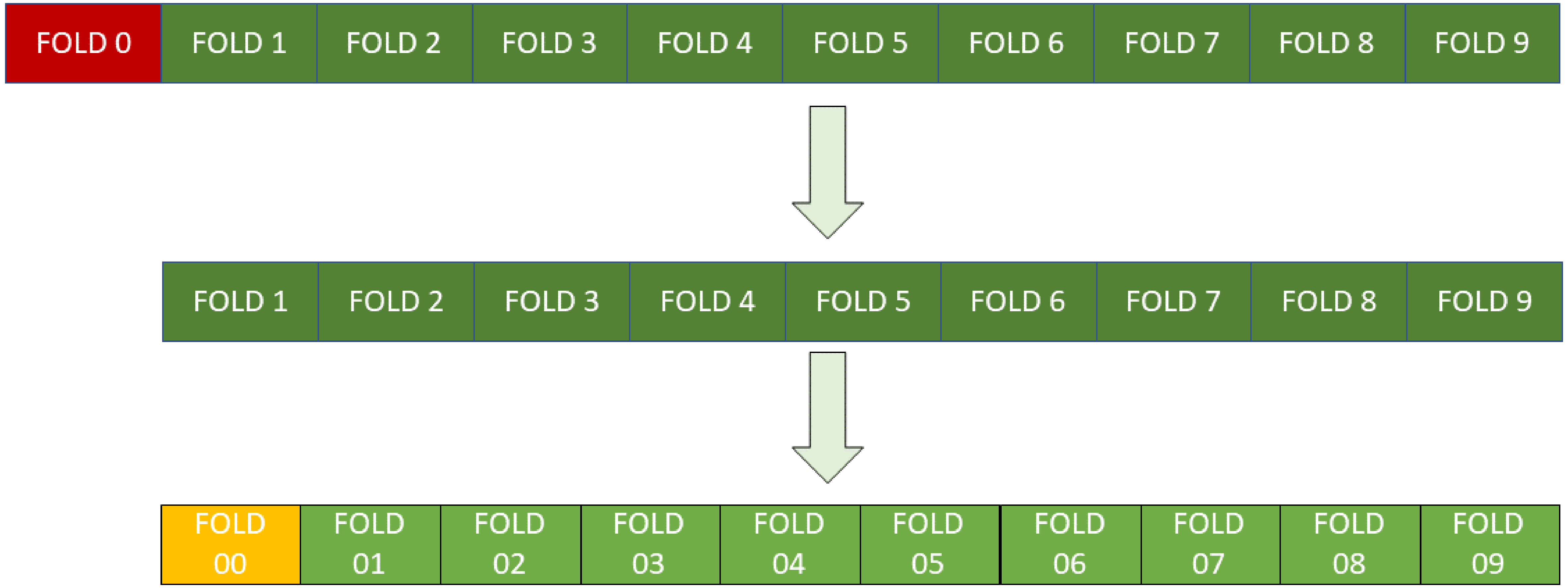


90%(Training)

80%

20%

CROSS-VALIDATION



FOLD 0	FOLD 1	FOLD 2	FOLD 3	FOLD 4	FOLD 5	FOLD 6	FOLD 7	FOLD 8	FOLD 9

FOLD 1	FOLD 2	FOLD 3	FOLD 4	FOLD 5	FOLD 6	FOLD 7	FOLD 8	FOLD 9

FOLD 00	FOLD 01	FOLD 02	FOLD 03	FOLD 04	FOLD 05	FOLD 06	FOLD 07	FOLD 08	FOLD 09

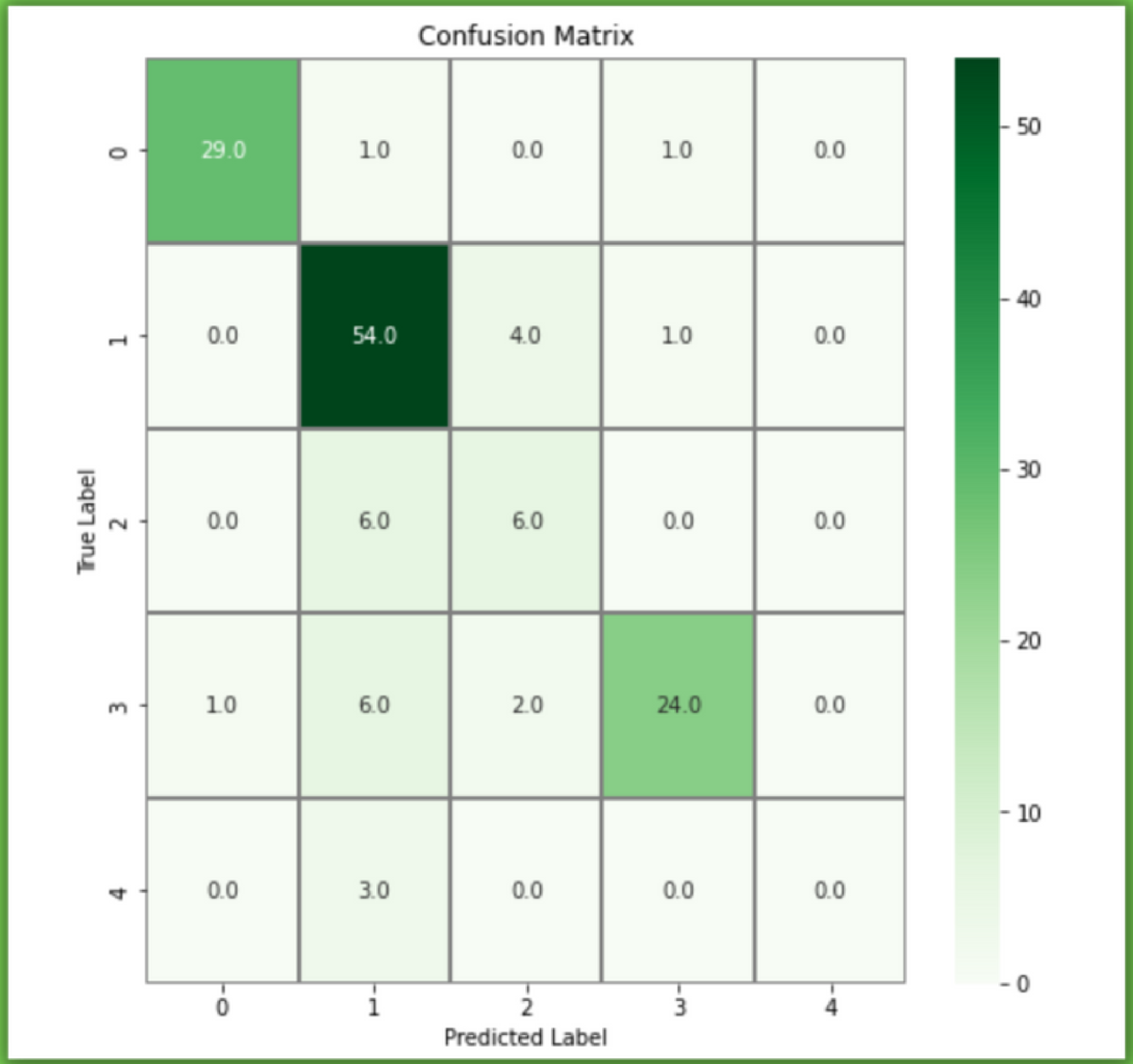
- LOSS AVERAGE: 0.495
- LOSS DEV.STD: 0.094
- ACCURACY AVERAGE: 0.815
- ACCURACY DEV.STD: 0.04

🟥 LOCAL TEST SET

🟩 Train

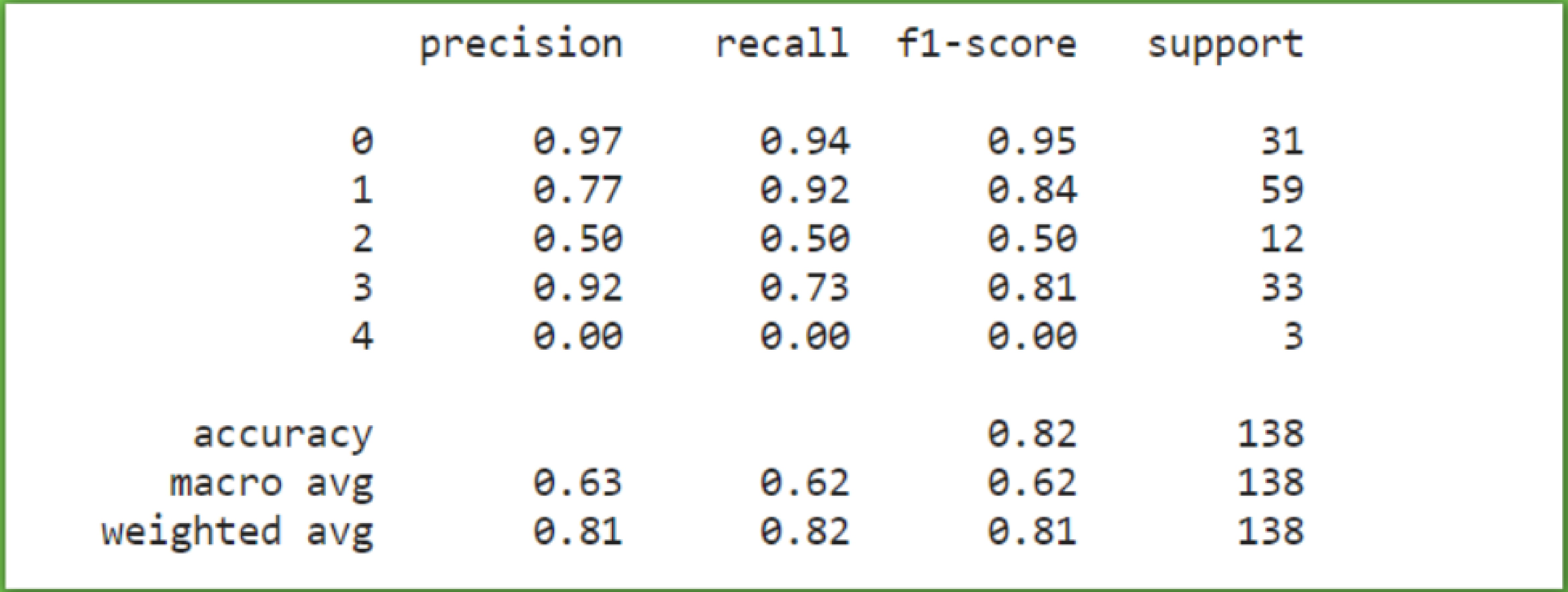
🟨 Validation

CONFUSION MATRIX AFTER RE-LEARNING



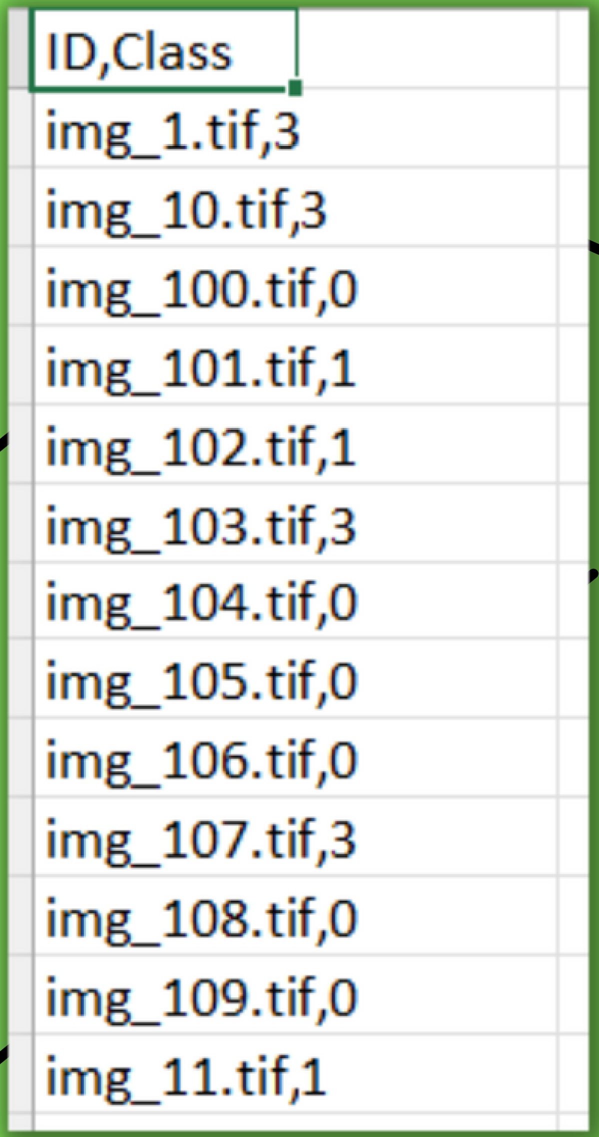
	0	1	2	3	4
0	29.0	1.0	0.0	1.0	0.0
1	0.0	54.0	4.0	1.0	0.0
2	0.0	6.0	6.0	0.0	0.0
3	1.0	6.0	2.0	24.0	0.0
4	0.0	3.0	0.0	0.0	0.0

PRECISION & RECALL



```
              precision    recall  f1-score   support

           0       0.97      0.94      0.95        31
           1       0.77      0.92      0.84        59
           2       0.50      0.50      0.50        12
           3       0.92      0.73      0.81        33
           4       0.00      0.00      0.00         3

    accuracy                           0.82       138
   macro avg       0.63      0.62      0.62       138
weighted avg       0.81      0.82      0.81       138
```

OUTPUT



```
ID,Class
img_1.tif,3
img_10.tif,3
img_100.tif,0
img_101.tif,1
img_102.tif,1
img_103.tif,3
img_104.tif,0
img_105.tif,0
img_106.tif,0
img_107.tif,3
img_108.tif,0
img_109.tif,0
img_11.tif,1
```