

# Histopathologic Cancer Detection

Hassan Youssef

Data-Science

Fazazi Zeid

Domin Wandrille

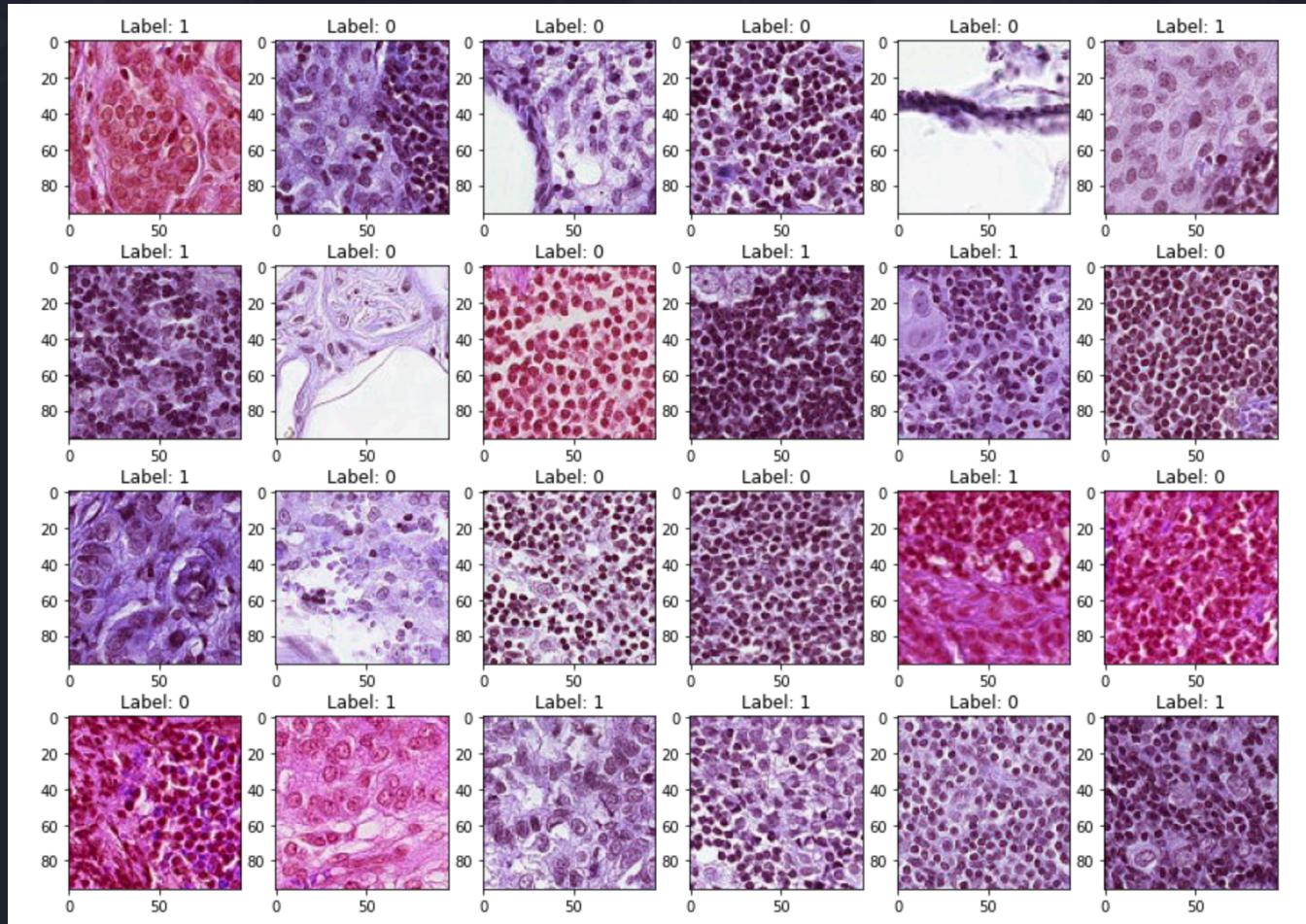
Mastier Lucas

L3 DANT

Aliev Rashid

2021/2022

# Présentation du dataset

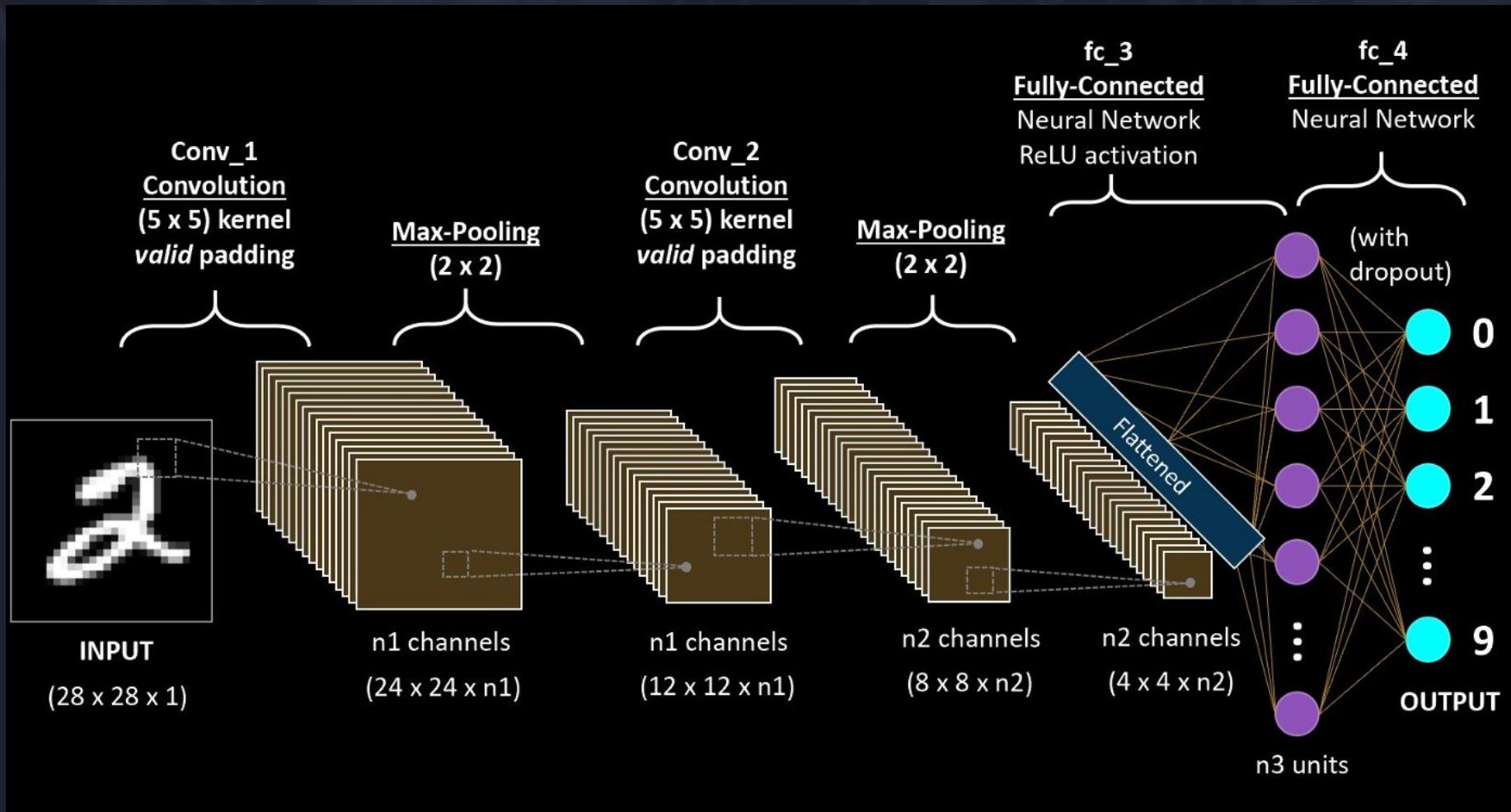


- ∨ dataset/histopathologic-cancer-detection
  - > test
  - > train
- sample\_submission.csv
- train\_labels.csv

## Train dataset

```
0    130908
1    89117
```

Name: label, dtype: int64



# Architectures

```

LR = 0.001
def build_compile_model1():
    model = Sequential()

    model.add(Conv2D(filters=256, kernel_size=(3,3), input_shape=(*IMG_SIZE, IMG_CHANNELS), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))

    model.add(Conv2D(filters=256, kernel_size=(3,3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))

    model.add(Flatten())

    model.add(Dense(64, activation='relu'))

    model.add(Dense(1))
    model.add(Activation('sigmoid'))

    model.compile(loss='binary_crossentropy',
                  optimizer=Adam(learning_rate=LR),
                  metrics=['accuracy'])

    return model

```

```

LR = 0.001
def build_compile_model2():
    kernel_size = (3,3)
    pool_size= (2,2)
    first_filters = 32
    second_filters = 64
    third_filters = 128
    dropout_conv = 0.3
    dropout_dense = 0.3

    # Model Structure
    model = Sequential()
    model.add(Conv2D(first_filters, kernel_size, activation='relu', input_shape=(*IMG_SIZE, IMG_CHANNELS)))
    model.add(Conv2D(first_filters, kernel_size, activation='relu'))
    model.add(Conv2D(first_filters, kernel_size, activation='relu'))
    model.add(MaxPooling2D(pool_size=pool_size))
    model.add(Dropout(dropout_conv))

    model.add(Conv2D(second_filters, kernel_size, activation='relu'))
    model.add(Conv2D(second_filters, kernel_size, activation='relu'))
    model.add(Conv2D(second_filters, kernel_size, activation='relu'))
    model.add(MaxPooling2D(pool_size=pool_size))
    model.add(Dropout(dropout_conv))

    model.add(Conv2D(third_filters, kernel_size, activation='relu'))
    model.add(Conv2D(third_filters, kernel_size, activation='relu'))
    model.add(Conv2D(third_filters, kernel_size, activation='relu'))
    model.add(MaxPooling2D(pool_size=pool_size))
    model.add(Dropout(dropout_conv))

    model.add(Flatten())
    model.add(Dense(256, activation="relu"))
    model.add(Dropout(dropout_dense))
    model.add(Dense(1, activation="sigmoid"))

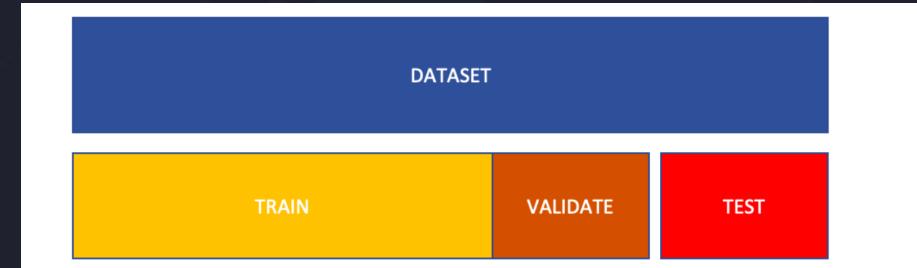
    model.compile(Adam(learning_rate=LR), loss='binary_crossentropy',
                  metrics=['accuracy'])

    return model

```

# Workflow

Charger le data set



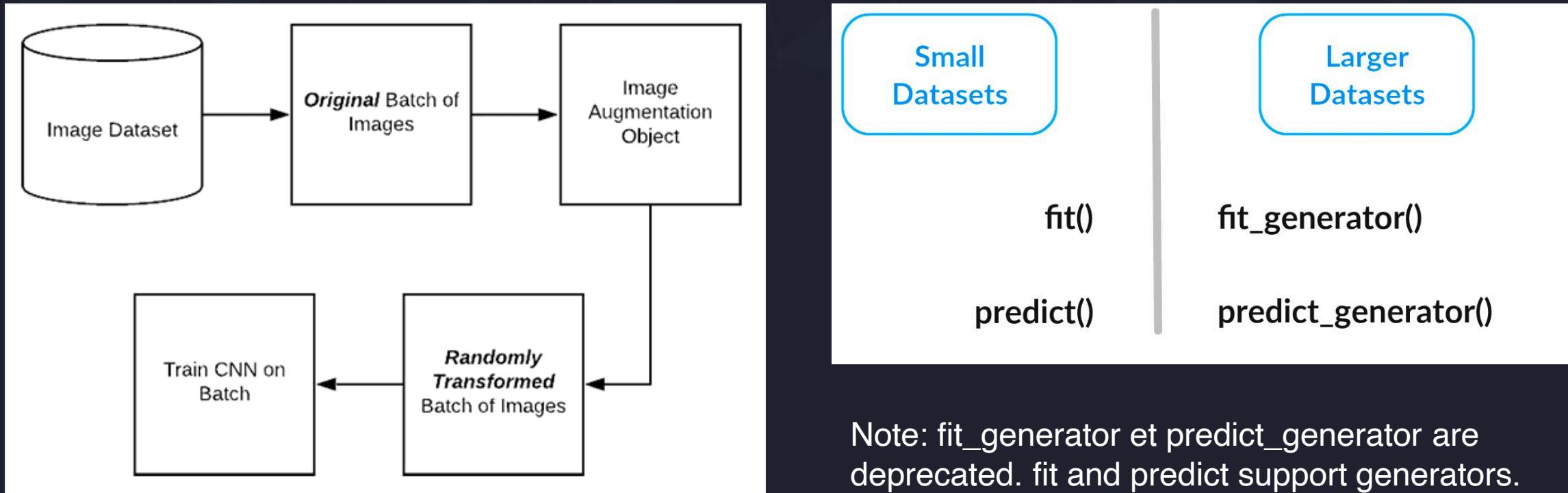
Préparer le data set

Entrainer les modèles (Forward & Back Propagation)

Tester les modèles (Forward Propagation)

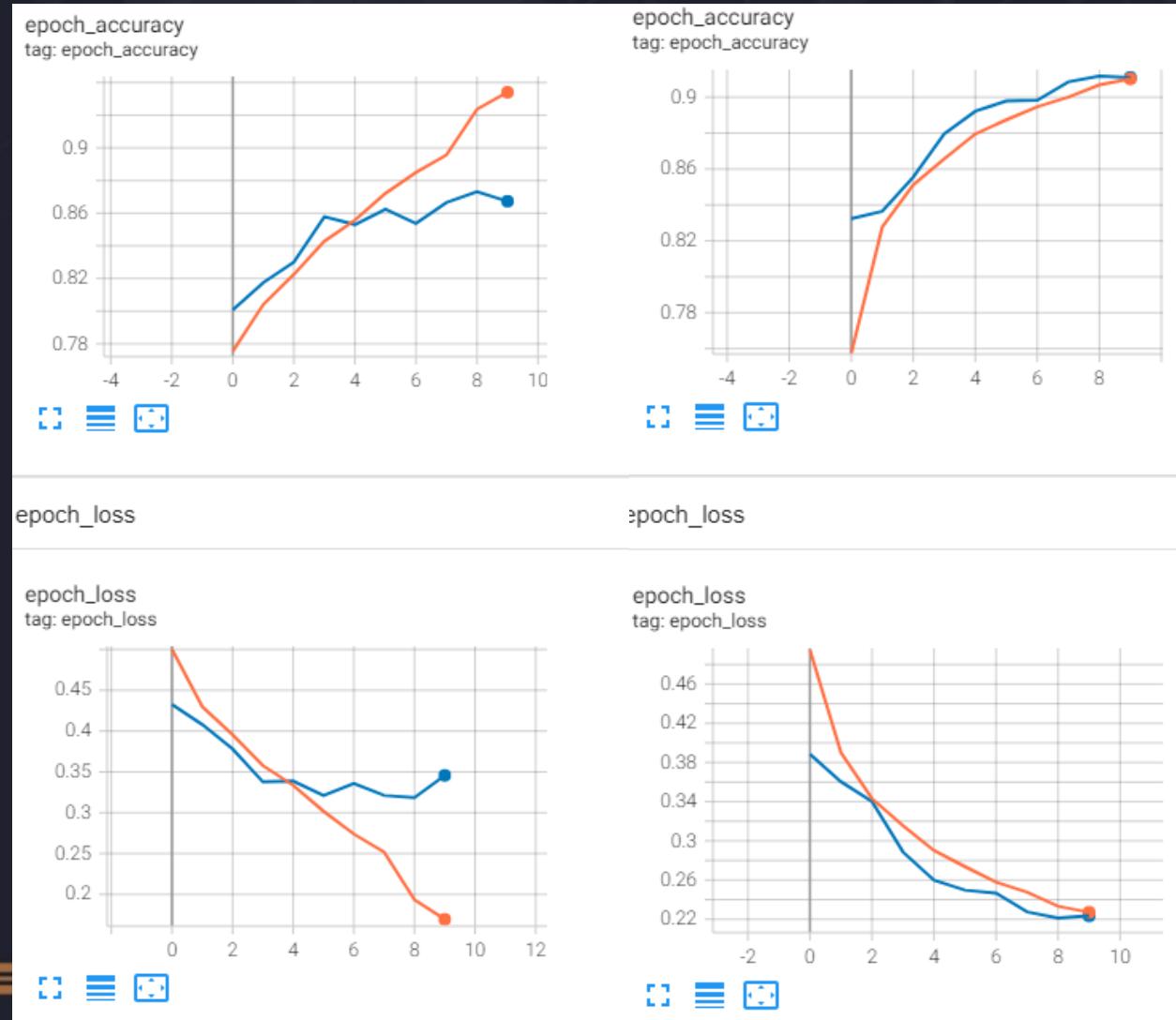
Analyser statistiquement les résultats prédictifs

# ImageDataGenerator



Note: `fit_generator` et `predict_generator` are deprecated. `fit` and `predict` support generators.

# TensorBoard

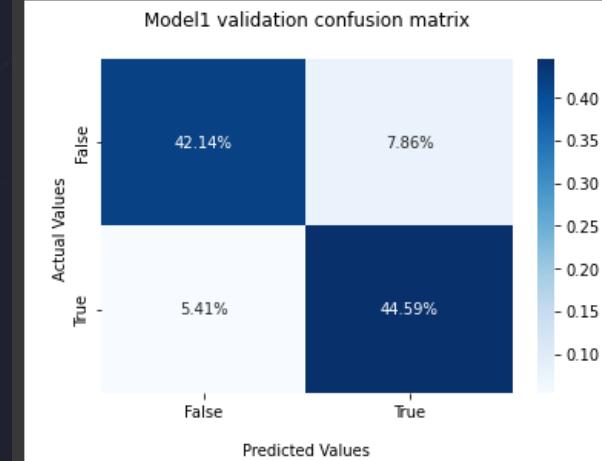


# Binary Classification Metrics

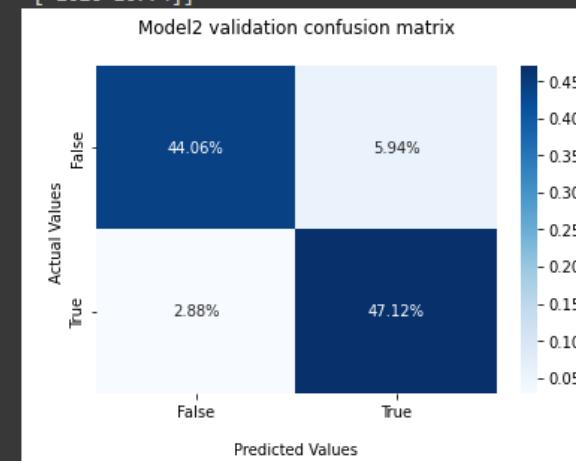
Metric	Formula
True positive rate, recall	$\frac{TP}{TP+FN}$
False positive rate	$\frac{FP}{FP+TN}$
Precision	$\frac{TP}{TP+FP}$
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
F-measure	$\frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$

# Confusion Matrix

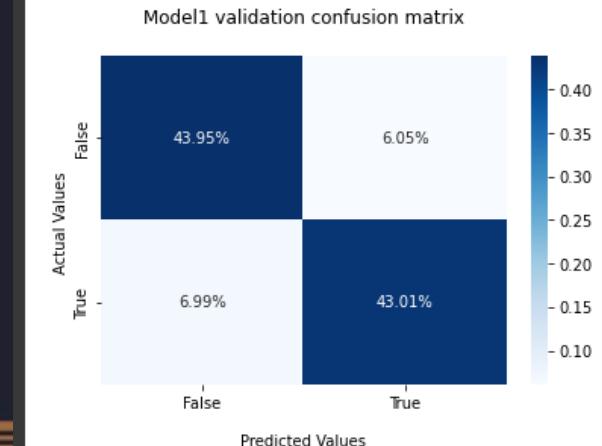
[[15002 2798]  
[ 1926 15874]]



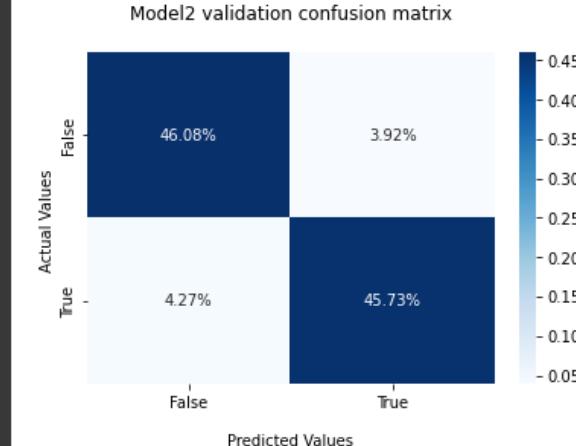
[[15685 2115]  
[ 1026 16774]]



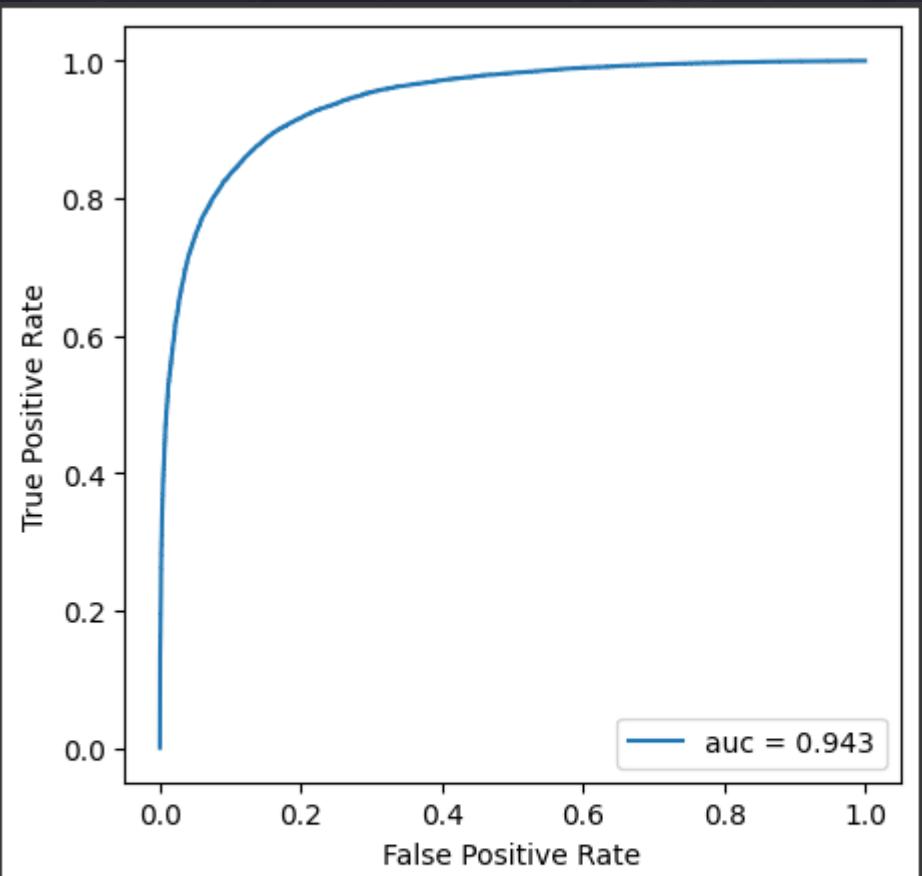
[[15645 2155]  
[ 2488 15312]]



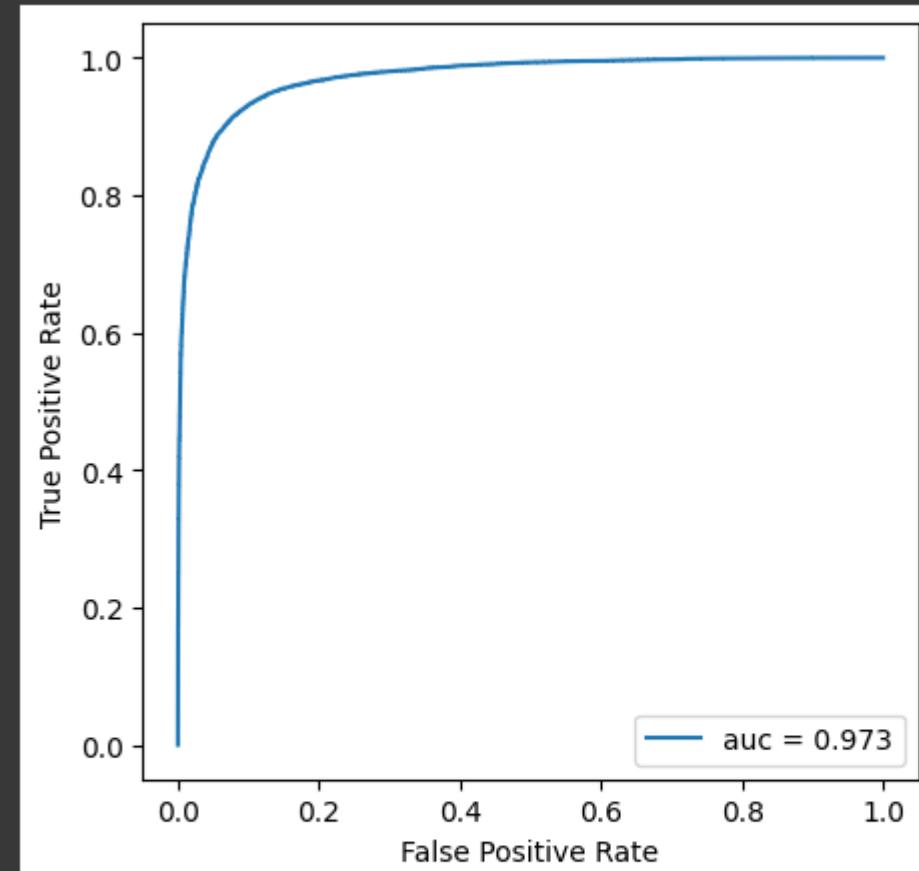
[[16405 1395]  
[ 1520 16280]]



## AUC-ROC Curve

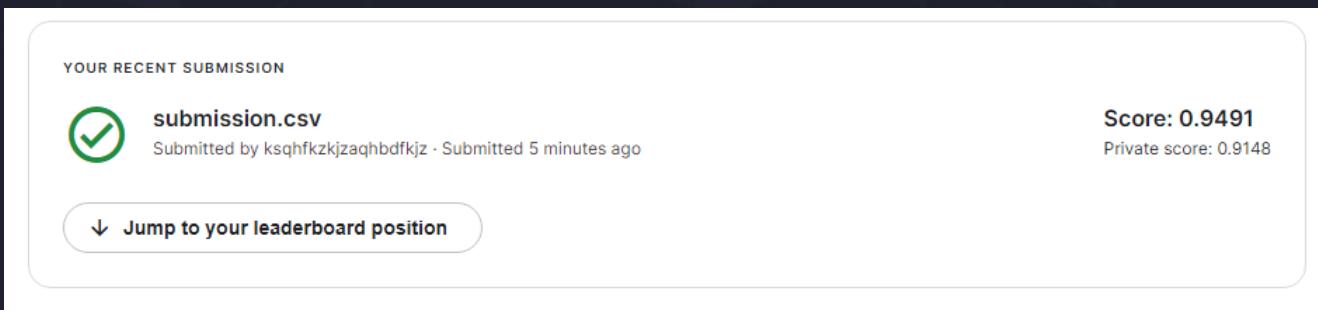


```
[0.          0.          0.          ... 0.98876404 0.98876404 1.          ]  
[0.0000000e+00 5.61797753e-05 2.80898876e-04 ... 9.99943820e-01  
1.0000000e+00 1.0000000e+00]  
[1.9999968e+00 9.9999678e-01 9.9999499e-01 ... 1.1485952e-05 1.1462010e-05  
3.4040716e-14]  
BEST THRESHOLD: 0.65364665
```



```
[0.          0.          0.          ... 0.97179775 0.97191011 1.          ]  
[0.0000000e+00 5.61797753e-05 2.24719101e-04 ... 1.0000000e+00  
1.0000000e+00 1.0000000e+00]  
[1.9999986e+00 9.9999857e-01 9.9999774e-01 ... 2.2486486e-03 2.2478900e-03  
6.2224061e-09]  
BEST THRESHOLD: 0.64063287
```

# Model 2 performance & tuning



	Class 1	Class 0
Breast	9/10	8/10
Lung	9/10	10/10
Colon	6/10	3/10

```
Results summary
Results in ./untitled_project
Showing 10 best trials
<keras_tuner.engine.objective.
Trial summary
Hyperparameters:
LR: 0.001
dropout_conv: 0.2
dropout_dense: 0.3
n_layers: 3
Score: 0.19306489825248718
Trial summary
Hyperparameters:
LR: 0.001
dropout_conv: 0.3
dropout_dense: 0.2
n_layers: 3
Score: 0.20378223061561584
Trial summary
Hyperparameters:
LR: 0.0005
dropout_conv: 0.3
dropout_dense: 0.2
n_layers: 4
Score: 0.32328590750694275
Trial summary
Hyperparameters:
LR: 0.0005
dropout_conv: 0.3
dropout_dense: 0.3
n_layers: 4
Score: 0.6931468844413757
Trial summary
Hyperparameters:
LR: 0.001
dropout_conv: 0.2
dropout_dense: 0.3
n_layers: 4
Score: 0.6931471228599548
```