# DenseNet121 - TB

October 27, 2023

[ ]:

```python
# importing packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import datetime

import warnings
warnings.simplefilter("ignore")

import tensorflow as tf
from keras import Model
from keras.models import Model
from keras.models import load_model, save_model
from keras.layers import Dense, Dropout, Flatten, AveragePooling2D
from keras.preprocessing.image import ImageDataGenerator
from keras.optimizers import Adam
from keras.callbacks import ModelCheckpoint, CSVLogger, ReduceLROnPlateau, TensorBoard
from keras.metrics import AUC, Precision, Recall
from keras.applications import DenseNet121

from sklearn.metrics import confusion_matrix, roc_curve, auc, precision_recall_curve
from sklearn.utils import class_weight

import os

from utils import *

%reload_ext autoreload
%autoreload 2
```

[ ]:

```python
# defining parameters
IMAGE_SIZE = 224
BATCH_SIZE = 32
```

```
INIT_LR = 0.001
EPOCHS = 20
```

## 0.1 Set up the file directory and image paths

[ ]:

```
# Set path for training testing and validation
# Data Generator for training, validation, and testing

folder = '../data/augmented_sorted'

# assigning paths
src_path_train = os.path.join(folder, 'train/')
src_path_val = os.path.join(folder, "val/")
src_path_test = os.path.join(folder,  'test/')

train_generator = ImageDataGenerator(rescale=1./255).flow_from_directory(
        src_path_train,
        target_size = (IMAGE_SIZE, IMAGE_SIZE),
        batch_size = BATCH_SIZE,
        class_mode = 'binary',
        shuffle = True,
        seed = 42)
valid_generator = ImageDataGenerator(rescale=1./255).flow_from_directory(
        src_path_val,
        target_size = (IMAGE_SIZE, IMAGE_SIZE),
        batch_size = BATCH_SIZE,
        class_mode = 'binary',
        shuffle = True,
        seed = 42)
test_generator = ImageDataGenerator(rescale=1./255).flow_from_directory(
        src_path_test,
        target_size = (IMAGE_SIZE, IMAGE_SIZE),
        batch_size = 1,
        class_mode = None,
        shuffle = False,
        seed = 42)

STEP_SIZE_TRAIN = train_generator.n // train_generator.batch_size
STEP_SIZE_VALID = valid_generator.n // valid_generator.batch_size
STEP_SIZE_TEST = test_generator.n // test_generator.batch_size
```

```
Found 10134 images belonging to 2 classes.
Found 1031 images belonging to 2 classes.
Found 1035 images belonging to 2 classes.
```

[ ]:

```
# calculate class weights
total_tb = len(os.listdir(os.path.join(src_path_train, 'TB')))
```

```
total_healthy = len(os.listdir(os.path.join(src_path_train, 'Non-TB')))

weight_for_0 = total_tb / (total_healthy + total_tb)
weight_for_1 = total_healthy / (total_healthy + total_tb)

class_weights = {0: weight_for_0, 1: weight_for_1}
class_weights
```

[ ]:

```
{0: 0.36895598973751725, 1: 0.6310440102624827}
```

[ ]:

```
# create model
def build_DenseNet_model():
    baseModel = DenseNet121(weights="imagenet",
                            include_top=False,
                            input_tensor=Input(shape=(IMAGE_SIZE, IMAGE_SIZE,␣
 ↪3)))
    output = baseModel.output
    output = AveragePooling2D(pool_size=(2, 2))(output)
    output = Flatten(name="flatten")(output)
    output = Dense(512, activation="relu")(output)
    output = Dropout(0.25)(output)
    output = Dense(1, activation="sigmoid")(output)
    model = Model(inputs=baseModel.input, outputs=output)
    for layer in baseModel.layers:
        layer.trainable = False
    return model


model = build_DenseNet_model()

MODEL_NAME = 'DenseNet'
model_path = os.path.join("logs", MODEL_NAME)

try:
    if not os.path.exists(model_path):
        os.makedirs(model_path)
        print("Model Directory Created")
    else:
        files = glob(os.path.join(model_path, "*"))
        for f in files:
            os.remove(f)
        print("Model Directory Already Exists and Files Deleted")
except Exception as e:
    print('Failed to delete %s. Reason: %s' % (model_path, e))


# compile model
```

```
model.compile(
    loss="binary_crossentropy",
    optimizer=tf.keras.optimizers.legacy.Adam(
        learning_rate=INIT_LR, decay=INIT_LR / EPOCHS
    ),
    metrics=["accuracy", AUC(), AUC(curve="PR"), Precision(), Recall()],
)
```

[ ]:

```
# define callback
# es = EarlyStopping(monitor = 'val_loss',
    #               verbose = 1,
    #               mode = 'min',
    #               patience = 4),
rlrp = ReduceLROnPlateau(monitor="val_accuracy",
                      patience=3,
                      verbose=1,
                      factor=0.5,
                      min_lr=0.00001)
mcp = ModelCheckpoint(filepath=os.path.join(model_path, "checkpoints/
 ↪model_{epoch:02d}_{val_loss:.2f}.hdf5"),
                    verbose=1,
                    monitor="val_loss",
                    mode="min")
cl = CSVLogger(os.path.join(model_path, "log.csv"))
tb = TensorBoard(log_dir=os.path.join(model_path, "tb_log"),
                 histogram_freq=1,
                 update_freq="batch")

# train model
history = model.fit(train_generator,
                    steps_per_epoch = STEP_SIZE_TRAIN,
                    validation_data = valid_generator,
                    validation_steps = STEP_SIZE_VALID,
                    epochs = EPOCHS,
                    class_weight = class_weights,
                    callbacks = [rlrp, mcp, cl, tb])
```

```
Model Directory Already Exists
```

[ ]:

```
# train model
history = model.fit(train_generator,
                    steps_per_epoch = STEP_SIZE_TRAIN,
                    validation_data = valid_generator,
                    validation_steps = STEP_SIZE_VALID,
                    epochs = EPOCHS,
                    class_weight = class_weights,
                     callbacks = [mcp_save, lr_reduction, csv_logger,␣
↪tensorboard_cb])
```

```
Epoch 1/20
633/633 [==============================] - ETA: 0s - loss: 0.1256 - accuracy:
0.9067 - auc: 0.9623 - auc_1: 0.9457 - precision: 0.8608 - recall: 0.8912
Epoch 1: saving model to logs/log_densenet/model.01-0.15.hdf5
633/633 [==============================] - 362s 565ms/step - loss: 0.1256 -
accuracy: 0.9067 - auc: 0.9623 - auc_1: 0.9457 - precision: 0.8608 - recall:
0.8912 - val_loss: 0.1487 - val_accuracy: 0.9502 - val_auc: 0.9831 - val_auc_1:
0.9583 - val_precision: 0.8719 - val_recall: 0.9134 - lr: 0.0010
Epoch 2/20
633/633 [==============================] - ETA: 0s - loss: 0.0651 - accuracy:
0.9445 - auc: 0.9877 - auc_1: 0.9822 - precision: 0.9150 - recall: 0.9366
Epoch 2: saving model to logs/log_densenet/model.02-0.15.hdf5
633/633 [==============================] - 354s 559ms/step - loss: 0.0651 -
accuracy: 0.9445 - auc: 0.9877 - auc_1: 0.9822 - precision: 0.9150 - recall:
0.9366 - val_loss: 0.1511 - val_accuracy: 0.9414 - val_auc: 0.9813 - val_auc_1:
0.9594 - val_precision: 0.9340 - val_recall: 0.7965 - lr: 0.0010
Epoch 3/20
633/633 [==============================] - ETA: 0s - loss: 0.0523 - accuracy:
0.9572 - auc: 0.9918 - auc_1: 0.9881 - precision: 0.9349 - recall: 0.9502
Epoch 3: saving model to logs/log_densenet/model.03-0.12.hdf5
633/633 [==============================] - 327s 517ms/step - loss: 0.0523 -
accuracy: 0.9572 - auc: 0.9918 - auc_1: 0.9881 - precision: 0.9349 - recall:
0.9502 - val_loss: 0.1167 - val_accuracy: 0.9570 - val_auc: 0.9872 - val_auc_1:
0.9701 - val_precision: 0.8816 - val_recall: 0.9351 - lr: 0.0010
Epoch 4/20
633/633 [==============================] - ETA: 0s - loss: 0.0445 - accuracy:
0.9618 - auc: 0.9939 - auc_1: 0.9907 - precision: 0.9391 - recall: 0.9585
Epoch 4: saving model to logs/log_densenet/model.04-0.10.hdf5
633/633 [==============================] - 329s 519ms/step - loss: 0.0445 -
accuracy: 0.9618 - auc: 0.9939 - auc_1: 0.9907 - precision: 0.9391 - recall:
0.9585 - val_loss: 0.1025 - val_accuracy: 0.9619 - val_auc: 0.9879 - val_auc_1:
0.9736 - val_precision: 0.9283 - val_recall: 0.9000 - lr: 0.0010
Epoch 5/20
633/633 [==============================] - ETA: 0s - loss: 0.0415 - accuracy:
0.9649 - auc: 0.9948 - auc_1: 0.9920 - precision: 0.9440 - recall: 0.9620
Epoch 5: saving model to logs/log_densenet/model.05-0.10.hdf5
633/633 [==============================] - 332s 524ms/step - loss: 0.0415 -
accuracy: 0.9649 - auc: 0.9948 - auc_1: 0.9920 - precision: 0.9440 - recall:
0.9620 - val_loss: 0.1003 - val_accuracy: 0.9639 - val_auc: 0.9896 - val_auc_1:
0.9754 - val_precision: 0.9079 - val_recall: 0.9353 - lr: 0.0010
Epoch 6/20
633/633 [==============================] - ETA: 0s - loss: 0.0317 - accuracy:
0.9740 - auc: 0.9969 - auc_1: 0.9944 - precision: 0.9580 - recall: 0.9722
Epoch 6: saving model to logs/log_densenet/model.06-0.11.hdf5
633/633 [==============================] - 332s 524ms/step - loss: 0.0317 -
```

```
accuracy: 0.9740 - auc: 0.9969 - auc_1: 0.9944 - precision: 0.9580 - recall:
0.9722 - val_loss: 0.1092 - val_accuracy: 0.9648 - val_auc: 0.9887 - val_auc_1:
0.9746 - val_precision: 0.9493 - val_recall: 0.8918 - lr: 0.0010
Epoch 7/20
633/633 [==============================] - ETA: 0s - loss: 0.0262 - accuracy:
0.9774 - auc: 0.9978 - auc_1: 0.9960 - precision: 0.9620 - recall: 0.9772
Epoch 7: saving model to logs/log_densenet/model.07-0.09.hdf5
633/633 [==============================] - 339s 535ms/step - loss: 0.0262 -
accuracy: 0.9774 - auc: 0.9978 - auc_1: 0.9960 - precision: 0.9620 - recall:
0.9772 - val_loss: 0.0899 - val_accuracy: 0.9629 - val_auc: 0.9914 - val_auc_1:
0.9750 - val_precision: 0.9106 - val_recall: 0.9264 - lr: 0.0010
Epoch 8/20
633/633 [==============================] - ETA: 0s - loss: 0.0256 - accuracy:
0.9780 - auc: 0.9979 - auc_1: 0.9965 - precision: 0.9641 - recall: 0.9767
Epoch 8: saving model to logs/log_densenet/model.08-0.14.hdf5
633/633 [==============================] - 349s 551ms/step - loss: 0.0256 -
accuracy: 0.9780 - auc: 0.9979 - auc_1: 0.9965 - precision: 0.9641 - recall:
0.9767 - val_loss: 0.1369 - val_accuracy: 0.9561 - val_auc: 0.9896 - val_auc_1:
0.9669 - val_precision: 0.8577 - val_recall: 0.9654 - lr: 0.0010
Epoch 9/20
633/633 [==============================] - ETA: 0s - loss: 0.0244 - accuracy:
0.9807 - auc: 0.9979 - auc_1: 0.9962 - precision: 0.9695 - recall: 0.9786
Epoch 9: saving model to logs/log_densenet/model.09-0.11.hdf5

Epoch 9: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.
633/633 [==============================] - 341s 539ms/step - loss: 0.0244 -
accuracy: 0.9807 - auc: 0.9979 - auc_1: 0.9962 - precision: 0.9695 - recall:
0.9786 - val_loss: 0.1134 - val_accuracy: 0.9600 - val_auc: 0.9879 - val_auc_1:
0.9759 - val_precision: 0.9567 - val_recall: 0.8615 - lr: 0.0010
Epoch 10/20
633/633 [==============================] - ETA: 0s - loss: 0.0134 - accuracy:
0.9895 - auc: 0.9995 - auc_1: 0.9989 - precision: 0.9837 - recall: 0.9879
Epoch 10: saving model to logs/log_densenet/model.10-0.10.hdf5
633/633 [==============================] - 354s 559ms/step - loss: 0.0134 -
accuracy: 0.9895 - auc: 0.9995 - auc_1: 0.9989 - precision: 0.9837 - recall:
0.9879 - val_loss: 0.1030 - val_accuracy: 0.9668 - val_auc: 0.9914 - val_auc_1:
0.9733 - val_precision: 0.9020 - val_recall: 0.9567 - lr: 5.0000e-04
Epoch 11/20
633/633 [==============================] - ETA: 0s - loss: 0.0102 - accuracy:
0.9917 - auc: 0.9997 - auc_1: 0.9993 - precision: 0.9880 - recall: 0.9896
Epoch 11: saving model to logs/log_densenet/model.11-0.10.hdf5
633/633 [==============================] - 336s 530ms/step - loss: 0.0102 -
accuracy: 0.9917 - auc: 0.9997 - auc_1: 0.9993 - precision: 0.9880 - recall:
0.9896 - val_loss: 0.0980 - val_accuracy: 0.9697 - val_auc: 0.9899 - val_auc_1:
0.9820 - val_precision: 0.9718 - val_recall: 0.8922 - lr: 5.0000e-04
Epoch 12/20
633/633 [==============================] - ETA: 0s - loss: 0.0097 - accuracy:
0.9924 - auc: 0.9998 - auc_1: 0.9996 - precision: 0.9880 - recall: 0.9914
```

```
Epoch 12: saving model to logs/log_densenet/model.12-0.10.hdf5
633/633 [==============================] - 353s 556ms/step - loss: 0.0097 -
accuracy: 0.9924 - auc: 0.9998 - auc_1: 0.9996 - precision: 0.9880 - recall:
0.9914 - val_loss: 0.0995 - val_accuracy: 0.9707 - val_auc: 0.9886 - val_auc_1:
0.9723 - val_precision: 0.9427 - val_recall: 0.9264 - lr: 5.0000e-04
Epoch 13/20
633/633 [==============================] - ETA: 0s - loss: 0.0091 - accuracy:
0.9922 - auc: 0.9997 - auc_1: 0.9994 - precision: 0.9872 - recall: 0.9917
Epoch 13: saving model to logs/log_densenet/model.13-0.12.hdf5
633/633 [==============================] - 336s 531ms/step - loss: 0.0091 -
accuracy: 0.9922 - auc: 0.9997 - auc_1: 0.9994 - precision: 0.9872 - recall:
0.9917 - val_loss: 0.1159 - val_accuracy: 0.9688 - val_auc: 0.9903 - val_auc_1:
0.9741 - val_precision: 0.9032 - val_recall: 0.9655 - lr: 5.0000e-04
Epoch 14/20
633/633 [==============================] - ETA: 0s - loss: 0.0064 - accuracy:
0.9955 - auc: 0.9999 - auc_1: 0.9998 - precision: 0.9933 - recall: 0.9944
Epoch 14: saving model to logs/log_densenet/model.14-0.13.hdf5
633/633 [==============================] - 339s 536ms/step - loss: 0.0064 -
accuracy: 0.9955 - auc: 0.9999 - auc_1: 0.9998 - precision: 0.9933 - recall:
0.9944 - val_loss: 0.1262 - val_accuracy: 0.9678 - val_auc: 0.9878 - val_auc_1:
0.9677 - val_precision: 0.9046 - val_recall: 0.9561 - lr: 5.0000e-04
Epoch 15/20
633/633 [==============================] - ETA: 0s - loss: 0.0057 - accuracy:
0.9951 - auc: 0.9999 - auc_1: 0.9999 - precision: 0.9915 - recall: 0.9952
Epoch 15: saving model to logs/log_densenet/model.15-0.12.hdf5

Epoch 15: ReduceLROnPlateau reducing learning rate to 0.0002500000118743628.
633/633 [==============================] - 343s 541ms/step - loss: 0.0057 -
accuracy: 0.9951 - auc: 0.9999 - auc_1: 0.9999 - precision: 0.9915 - recall:
0.9952 - val_loss: 0.1233 - val_accuracy: 0.9658 - val_auc: 0.9863 - val_auc_1:
0.9722 - val_precision: 0.9372 - val_recall: 0.9087 - lr: 5.0000e-04
Epoch 16/20
633/633 [==============================] - ETA: 0s - loss: 0.0033 - accuracy:
0.9983 - auc: 1.0000 - auc_1: 1.0000 - precision: 0.9979 - recall: 0.9976
Epoch 16: saving model to logs/log_densenet/model.16-0.11.hdf5
633/633 [==============================] - 347s 548ms/step - loss: 0.0033 -
accuracy: 0.9983 - auc: 1.0000 - auc_1: 1.0000 - precision: 0.9979 - recall:
0.9976 - val_loss: 0.1061 - val_accuracy: 0.9717 - val_auc: 0.9892 - val_auc_1:
0.9744 - val_precision: 0.9280 - val_recall: 0.9481 - lr: 2.5000e-04
Epoch 17/20
633/633 [==============================] - ETA: 0s - loss: 0.0034 - accuracy:
0.9977 - auc: 1.0000 - auc_1: 1.0000 - precision: 0.9960 - recall: 0.9979
Epoch 17: saving model to logs/log_densenet/model.17-0.12.hdf5
633/633 [==============================] - 344s 542ms/step - loss: 0.0034 -
accuracy: 0.9977 - auc: 1.0000 - auc_1: 1.0000 - precision: 0.9960 - recall:
0.9979 - val_loss: 0.1192 - val_accuracy: 0.9668 - val_auc: 0.9879 - val_auc_1:
0.9674 - val_precision: 0.9195 - val_recall: 0.9353 - lr: 2.5000e-04
Epoch 18/20
```

```
633/633 [==============================] - ETA: 0s - loss: 0.0029 - accuracy:
0.9977 - auc: 1.0000 - auc_1: 1.0000 - precision: 0.9963 - recall: 0.9976
Epoch 18: saving model to logs/log_densenet/model.18-0.12.hdf5
633/633 [==============================] - 346s 547ms/step - loss: 0.0029 -
accuracy: 0.9977 - auc: 1.0000 - auc_1: 1.0000 - precision: 0.9963 - recall:
0.9976 - val_loss: 0.1242 - val_accuracy: 0.9707 - val_auc: 0.9856 - val_auc_1:
0.9621 - val_precision: 0.9351 - val_recall: 0.9351 - lr: 2.5000e-04
Epoch 19/20
633/633 [==============================] - ETA: 0s - loss: 0.0029 - accuracy:
0.9981 - auc: 1.0000 - auc_1: 1.0000 - precision: 0.9971 - recall: 0.9979
Epoch 19: saving model to logs/log_densenet/model.19-0.15.hdf5

Epoch 19: ReduceLROnPlateau reducing learning rate to 0.0001250000059371814.
633/633 [==============================] - 342s 540ms/step - loss: 0.0029 -
accuracy: 0.9981 - auc: 1.0000 - auc_1: 1.0000 - precision: 0.9971 - recall:
0.9979 - val_loss: 0.1500 - val_accuracy: 0.9678 - val_auc: 0.9868 - val_auc_1:
0.9579 - val_precision: 0.8964 - val_recall: 0.9698 - lr: 2.5000e-04
Epoch 20/20
633/633 [==============================] - ETA: 0s - loss: 0.0019 - accuracy:
0.9989 - auc: 1.0000 - auc_1: 1.0000 - precision: 0.9981 - recall: 0.9989
Epoch 20: saving model to logs/log_densenet/model.20-0.13.hdf5
633/633 [==============================] - 339s 535ms/step - loss: 0.0019 -
accuracy: 0.9989 - auc: 1.0000 - auc_1: 1.0000 - precision: 0.9981 - recall:
0.9989 - val_loss: 0.1250 - val_accuracy: 0.9688 - val_auc: 0.9879 - val_auc_1:
0.9647 - val_precision: 0.9237 - val_recall: 0.9397 - lr: 1.2500e-04
```

[ ]:

```
save_model(
    model,
    os.path.join(model_path, "DenseNet-model.h5"),
    overwrite=True,
    include_optimizer=True,
    save_format=None,
    signatures=None,
    options=None,
)

model.save(os.path.join(model_path, "DenseNet-model.keras"))
model.save_weights(os.path.join(model_path, "DenseNet-weights.h5"))
```

[ ]:

```
history = pd.read_csv('../modeling/logs/log_densenet/log.csv')
history
```

[ ]:

```
   epoch  accuracy       auc     auc_1      loss        lr  precision  \
0      0  0.906701  0.962340  0.945699  0.125611  0.001000   0.860802
1      1  0.944455  0.987723  0.982215  0.065120  0.001000   0.915010
2      2  0.957205  0.991761  0.988055  0.052331  0.001000   0.934897
3      3  0.961751  0.993941  0.990686  0.044471  0.001000   0.939076
```

```
4       4  0.964914  0.994751  0.991971  0.041460  0.001000   0.943992
5       5  0.974007  0.996855  0.994440  0.031731  0.001000   0.958048
6       6  0.977367  0.997792  0.996037  0.026191  0.001000   0.962025
7       7  0.977960  0.997851  0.996501  0.025550  0.001000   0.964059
8       8  0.980727  0.997926  0.996187  0.024361  0.001000   0.969496
9       9  0.989524  0.999453  0.998934  0.013442  0.000500   0.983725
10     10  0.991698  0.999683  0.999315  0.010221  0.000500   0.987971
11     11  0.992390  0.999762  0.999601  0.009712  0.000500   0.987987
12     12  0.992192  0.999710  0.999364  0.009081  0.000500   0.987200
13     13  0.995454  0.999903  0.999837  0.006359  0.000500   0.993312
14     14  0.995058  0.999929  0.999880  0.005721  0.000500   0.991469
15     15  0.998320  0.999980  0.999966  0.003341  0.000250   0.997857
16     16  0.997727  0.999969  0.999950  0.003372  0.000250   0.995993
17     17  0.997727  0.999986  0.999977  0.002939  0.000250   0.996256
18     18  0.998122  0.999974  0.999957  0.002904  0.000250   0.997055
19     19  0.998913  0.999997  0.999994  0.001863  0.000125   0.998126

      recall  val_accuracy   val_auc  val_auc_1  val_loss  val_precision  \
0   0.891240      0.950195  0.983072   0.958303  0.148746       0.871901
1   0.936563      0.941406  0.981347   0.959384  0.151120       0.934010
2   0.950174      0.957031  0.987196   0.970079  0.116735       0.881633
3   0.958456      0.961914  0.987852   0.973631  0.102534       0.928251
4   0.961951      0.963867  0.989581   0.975364  0.100319       0.907950
5   0.972155      0.964844  0.988686   0.974554  0.109166       0.949309
6   0.977230      0.962891  0.991413   0.975030  0.089883       0.910638
7   0.976707      0.956055  0.989620   0.966939  0.136893       0.857692
8   0.978581      0.959961  0.987911   0.975862  0.113446       0.956731
9   0.987942      0.966797  0.991437   0.973296  0.102957       0.902041
10  0.989558      0.969727  0.989948   0.981984  0.098030       0.971831
11  0.991428      0.970703  0.988604   0.972262  0.099540       0.942731
12  0.991696      0.968750  0.990299   0.974052  0.115864       0.903226
13  0.994376      0.967773  0.987823   0.967668  0.126166       0.904564
14  0.995183      0.965820  0.986272   0.972221  0.123338       0.937220
15  0.997590      0.971680  0.989246   0.974431  0.106107       0.927966
16  0.997859      0.966797  0.987880   0.967391  0.119166       0.919492
17  0.997590      0.970703  0.985580   0.962120  0.124185       0.935065
18  0.997856      0.967773  0.986767   0.957902  0.150025       0.896414
19  0.998928      0.968750  0.987872   0.964749  0.125002       0.923729

    val_recall
0     0.913420
1     0.796537
2     0.935065
3     0.900000
4     0.935345
5     0.891775
6     0.926407
```

```
7     0.965368
8     0.861472
9     0.956710
10    0.892241
11    0.926407
12    0.965517
13    0.956140
14    0.908696
15    0.948052
16    0.935345
17    0.935065
18    0.969828
19    0.939655
```

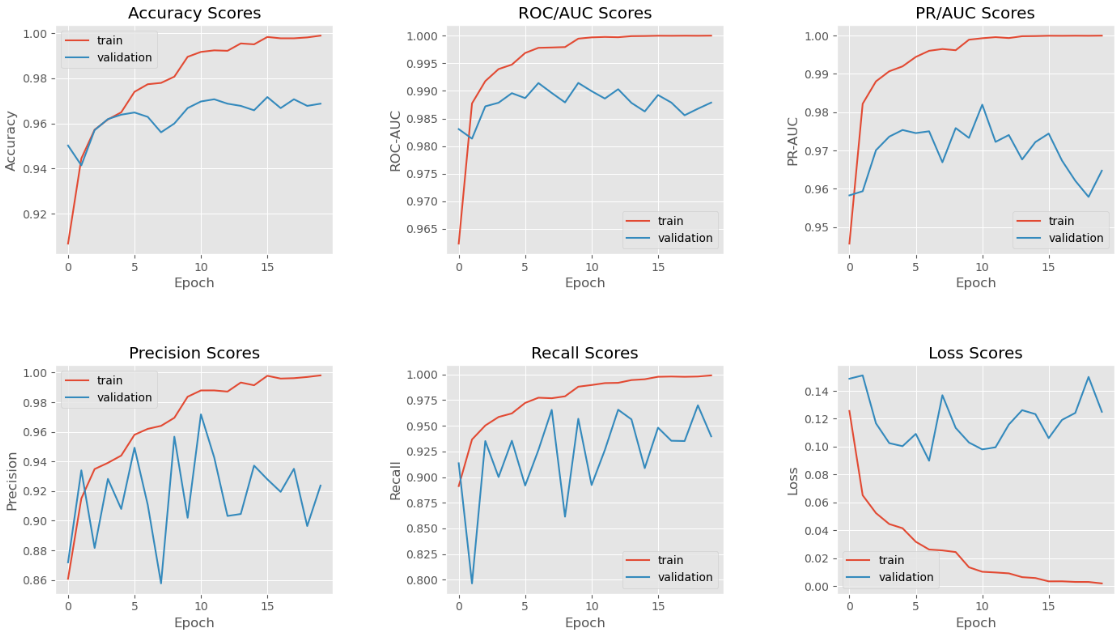
[ ]:

```
model = tf.keras.models.load_model('../modeling/logs/log_densenet/model.40-0.42.
  ↪hdf5')
```

```
2023-08-31 12:38:06.442945: I metal_plugin/src/device/metal_device.cc:1154]
Metal device set to: Apple M1 Pro
2023-08-31 12:38:06.442978: I metal_plugin/src/device/metal_device.cc:296]
systemMemory: 16.00 GB
2023-08-31 12:38:06.442991: I metal_plugin/src/device/metal_device.cc:313]
maxCacheSize: 5.33 GB
2023-08-31 12:38:06.443356: I
tensorflow/core/common_runtime/pluggable_device/pluggable_device_factory.cc:303]
Could not identify NUMA node of platform GPU ID 0, defaulting to 0. Your kernel
may not have been built with NUMA support.
2023-08-31 12:38:06.443659: I
tensorflow/core/common_runtime/pluggable_device/pluggable_device_factory.cc:269]
Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 0
MB memory) -> physical PluggableDevice (device: 0, name: METAL, pci bus id:
<undefined>)
```

[ ]:

```
# evaluate model by displaying metrics with visualization
plt.style.use('ggplot')
sixplot2(history, 'auc', 'val_auc', 'auc_1', 'val_auc_1', 'precision',␣
  ↪'val_precision', 'recall', 'val_recall')
```



[ ]:

```
score = model.evaluate_generator(generator=valid_generator,
                                  steps=STEP_SIZE_VALID)

print("Validation Loss: " + str(score[0]))
print("Validation Accuracy: " + str(score[1]))
print("Validation Precision: " + str(score[2]))
print("Validation Recall: " + str(score[3]))
print("Validation ROC-AUC: " + str(score[4]))
print("Validation PR-AUC: " + str(score[5]))
```

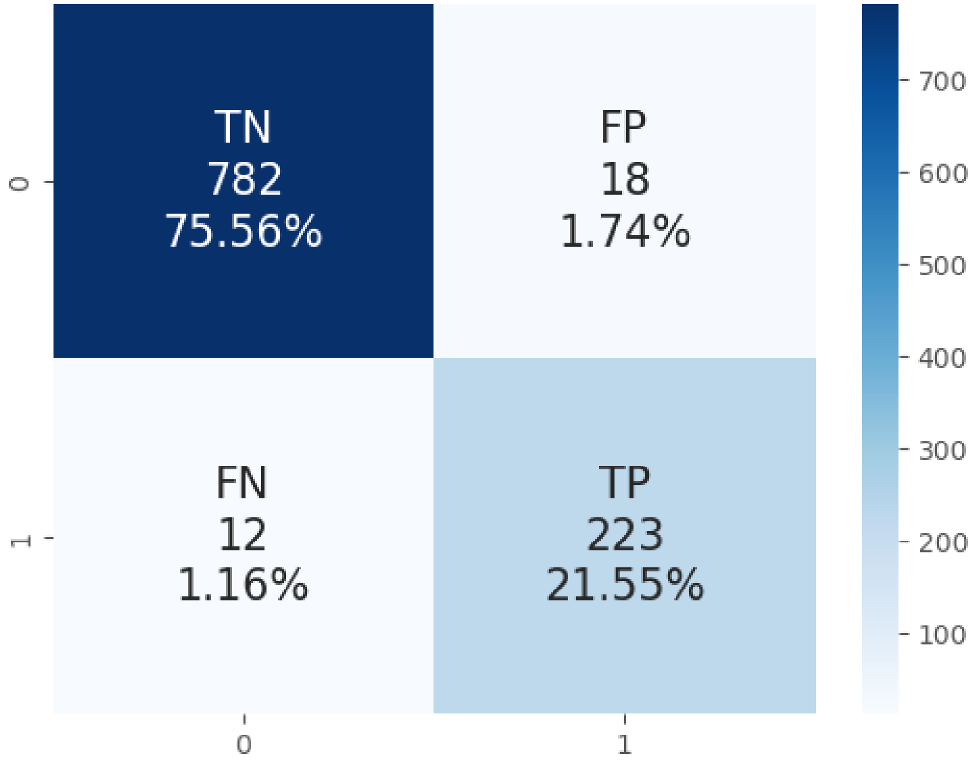
```
Validation Loss: 0.1250956505537033
Validation Accuracy: 0.96875
Validation Precision: 0.9877833724021912
Validation Recall: 0.9642906785011292
Validation ROC-AUC: 0.9230769276618958
Validation PR-AUC: 0.939130425453186
```

# 1 Testing

[ ]:

```
STEP_SIZE_TEST = test_generator.n // test_generator.batch_size

# get testing accuracy and loss
test_generator.reset()
# display confusion matrix visualization
pred = model.predict_generator(test_generator,
                               steps = STEP_SIZE_TEST,
                                                verbose = 1)
y_true = test_generator.classes
y_pred = pred > 0.5
make_confusion_matrix(y_true, y_pred)
```

```
1035/1035 [==============================] - 60s 58ms/step
```



[ ]:
```
predicted_class_indices = np.argmax(pred, axis=1)

labels = (train_generator.class_indices)
labels = dict((v,k) for k,v in labels.items())
predictions = [labels[k] for k in predicted_class_indices]
```

[ ]:
```
y_true
```

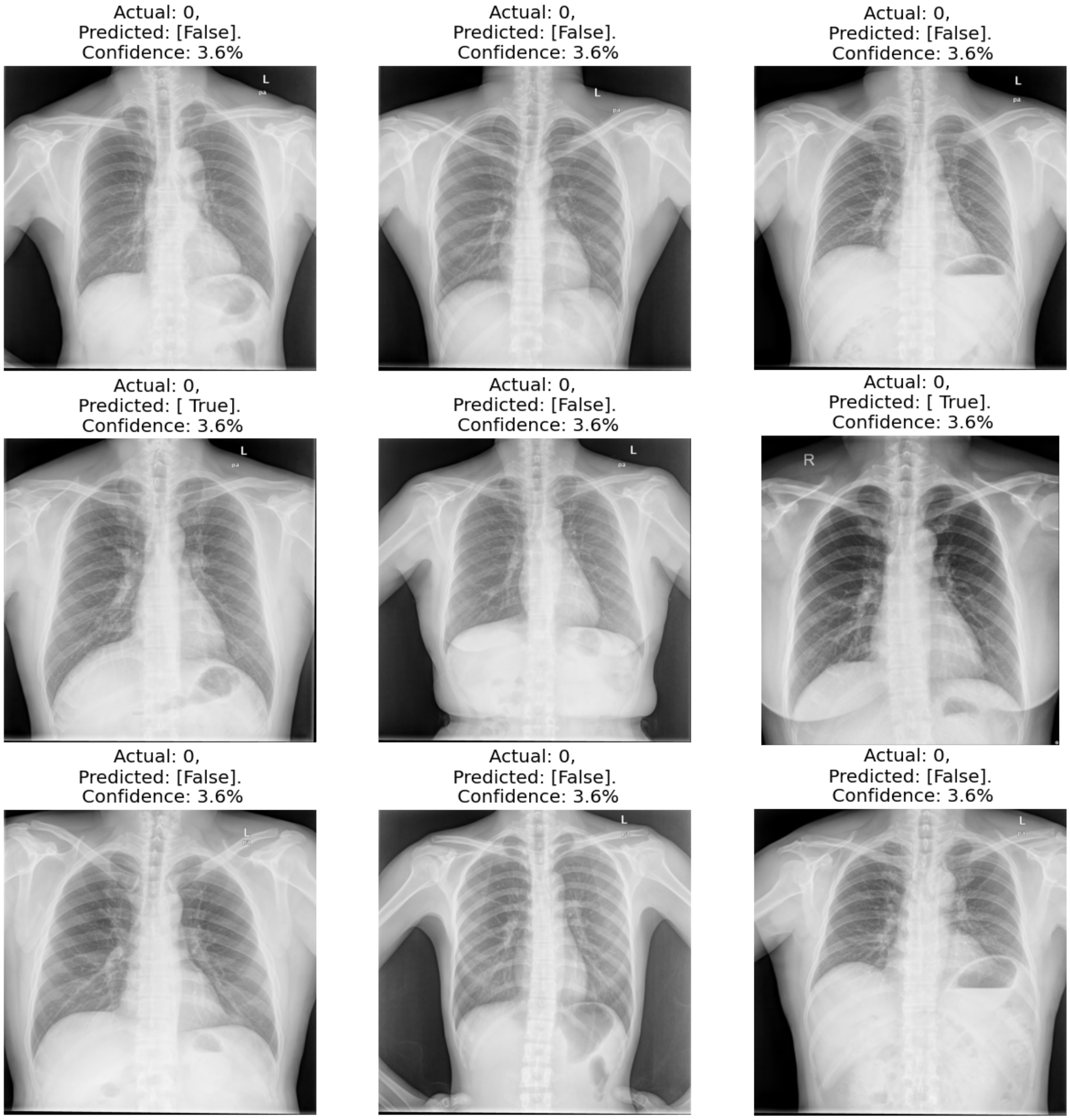
[ ]:
```
array([0, 0, 0, …, 1, 1, 1], dtype=int32)
```

[ ]:
```
labels
```

[ ]:
```
{0: 'Non-TB', 1: 'TB'}
```

[ ]:

```
plt.figure(figsize=(15,15))
for i in range(9):
    ax = plt.subplot(3,3,i+1)
    image = plt.imread(os.path.join('../data/augmented_sorted/test',␣
 ↪test_generator.filenames[i]))
    plt.imshow(image)

    predicted_class = y_pred[i]
    confidence = round(100 * (np.max(pred[0])),2)
    actual_class = y_true[i]

    plt.title(f"Actual: {actual_class}, \n Predicted: {predicted_class}. \n␣
 ↪Confidence: {confidence}%")
    plt.axis('off')
```



[ ]:

```
test_generator.filenames
```

[ ]:

```
['Non-TB/CHNCXR_0004_0.png',
 'Non-TB/CHNCXR_0005_0.png',
 'Non-TB/CHNCXR_0014_0.png',
 'Non-TB/CHNCXR_0018_0.png',
 'Non-TB/CHNCXR_0027_0.png',
 'Non-TB/CHNCXR_0030_0.png',
 'Non-TB/CHNCXR_0046_0.png',
 'Non-TB/CHNCXR_0054_0.png',
 'Non-TB/CHNCXR_0059_0.png',
 'Non-TB/CHNCXR_0065_0.png',
 'Non-TB/CHNCXR_0078_0.png',
 'Non-TB/CHNCXR_0081_0.png',
 'Non-TB/CHNCXR_0096_0.png',
 'Non-TB/CHNCXR_0122_0.png',
 'Non-TB/CHNCXR_0161_0.png',
 'Non-TB/CHNCXR_0174_0.png',
 'Non-TB/CHNCXR_0195_0.png',
 'Non-TB/CHNCXR_0202_0.png',
 'Non-TB/CHNCXR_0205_0.png',
 'Non-TB/CHNCXR_0206_0.png',
 'Non-TB/CHNCXR_0219_0.png',
 'Non-TB/CHNCXR_0228_0.png',
 'Non-TB/CHNCXR_0246_0.png',
 'Non-TB/CHNCXR_0259_0.png',
 'Non-TB/CHNCXR_0262_0.png',
 'Non-TB/CHNCXR_0264_0.png',
 'Non-TB/CHNCXR_0297_0.png',
 'Non-TB/CHNCXR_0315_0.png',
 'Non-TB/CHNCXR_0319_0.png',
 'Non-TB/MCUCXR_0001_0.png',
 'Non-TB/MCUCXR_0002_0.png',
 'Non-TB/MCUCXR_0024_0.png',
 'Non-TB/MCUCXR_0044_0.png',
 'Non-TB/MCUCXR_0051_0.png',
 'Non-TB/MCUCXR_0053_0.png',
 'Non-TB/MCUCXR_0068_0.png',
 'Non-TB/MCUCXR_0071_0.png',
 'Non-TB/MCUCXR_0081_0.png',
 'Non-TB/h0008.png',
 'Non-TB/h0013.png',
 'Non-TB/h0017.png',
 'Non-TB/h0048.png',
 'Non-TB/h0067.png',
 'Non-TB/h0075.png',
 'Non-TB/h0079.png',
 'Non-TB/h0085.png',
 'Non-TB/h0093.png',
 'Non-TB/h0094.png',
 'Non-TB/h0096.png',
 'Non-TB/h0114.png',
 'Non-TB/h0117.png',
 'Non-TB/h0142.png',
 'Non-TB/h0146.png',
 'Non-TB/h0167.png',
 'Non-TB/h0180.png',
 'Non-TB/h0186.png',
 'Non-TB/h0192.png',
 'Non-TB/h0194.png',
 'Non-TB/h0198.png',
 'Non-TB/h0201.png',
 'Non-TB/h0202.png',
 'Non-TB/h0209.png',
 'Non-TB/h0218.png',
 'Non-TB/h0219.png',
 'Non-TB/h0224.png',
 'Non-TB/h0237.png',
 'Non-TB/h0239.png',
 'Non-TB/h0244.png',
 'Non-TB/h0264.png',
 'Non-TB/h0265.png',
 'Non-TB/h0299.png',
 'Non-TB/h0310.png',
 'Non-TB/h0313.png',
 'Non-TB/h0329.png',
 'Non-TB/h0358.png',
 'Non-TB/h0368.png',
 'Non-TB/h0381.png',
 'Non-TB/h0420.png',
 'Non-TB/h0433.png',
 'Non-TB/h0436.png',
 'Non-TB/h0444.png',
 'Non-TB/h0445.png',
 'Non-TB/h0447.png',
 'Non-TB/h0453.png',
 'Non-TB/h0479.png',
 'Non-TB/h0481.png',
 'Non-TB/h0494.png',
 'Non-TB/h0500.png',
 'Non-TB/h0520.png',
 'Non-TB/h0533.png',
 'Non-TB/h0548.png',
 'Non-TB/h0550.png',
 'Non-TB/h0553.png',
 'Non-TB/h0561.png',
 'Non-TB/h0596.png',
 'Non-TB/h0598.png',
 'Non-TB/h0603.png',
 'Non-TB/h0617.png',
 'Non-TB/h0621.png',
 'Non-TB/h0635.png',
 'Non-TB/h0639.png',
 'Non-TB/h0656.png',
 'Non-TB/h0657.png',
 'Non-TB/h0661.png',
 'Non-TB/h0691.png',
 'Non-TB/h0698.png',
 'Non-TB/h0700.png',
 'Non-TB/h0702.png',
 'Non-TB/h0751.png',
 'Non-TB/h0766.png',
 'Non-TB/h0808.png',
 'Non-TB/h0811.png',
 'Non-TB/h0823.png',
 'Non-TB/h0840.png',
 'Non-TB/h0842.png',
 'Non-TB/h0851.png',
 'Non-TB/h0874.png',
 'Non-TB/h0875.png',
 'Non-TB/h0890.png',
 'Non-TB/h0892.png',
 'Non-TB/h0914.png',
 'Non-TB/h0955.png',
 'Non-TB/h0969.png',
 'Non-TB/h0975.png',
 'Non-TB/h1002.png',
 'Non-TB/h1046.png',
 'Non-TB/h1047.png',
 'Non-TB/h1052.png',
 'Non-TB/h1062.png',
 'Non-TB/h1103.png',
 'Non-TB/h1130.png',
 'Non-TB/h1133.png',
 'Non-TB/h1138.png',
 'Non-TB/h1159.png',
 'Non-TB/h1164.png',
 'Non-TB/h1169.png',
 'Non-TB/h1181.png',
 'Non-TB/h1184.png',
 'Non-TB/h1187.png',
 'Non-TB/h1204.png',
 'Non-TB/h1217.png',
 'Non-TB/h1218.png',
 'Non-TB/h1226.png',
 'Non-TB/h1239.png',
 'Non-TB/h1244.png',
 'Non-TB/h1255.png',
 'Non-TB/h1301.png',
 'Non-TB/h1314.png',
 'Non-TB/h1349.png',
 'Non-TB/h1366.png',
 'Non-TB/h1379.png',
 'Non-TB/h1394.png',
 'Non-TB/h1395.png',
 'Non-TB/h1405.png',
 'Non-TB/h1452.png',
 'Non-TB/h1478.png',
 'Non-TB/h1523.png',
 'Non-TB/h1525.png',
 'Non-TB/h1526.png',
 'Non-TB/h1528.png',
 'Non-TB/h1538.png',
 'Non-TB/h1540.png',
 'Non-TB/h1546.png',
 'Non-TB/h1556.png',
 'Non-TB/h1567.png',
 'Non-TB/h1600.png',
 'Non-TB/h1620.png',
 'Non-TB/h1621.png',
 'Non-TB/h1625.png',
 'Non-TB/h1628.png',
 'Non-TB/h1643.png',
 'Non-TB/h1665.png',
 'Non-TB/h1673.png',
 'Non-TB/h1727.png',
 'Non-TB/h1732.png',
 'Non-TB/h1733.png',
 'Non-TB/h1741.png',
 'Non-TB/h1744.png',
 'Non-TB/h1755.png',
 'Non-TB/h1763.png',
 'Non-TB/h1769.png',
 'Non-TB/h1773.png',
 'Non-TB/h1774.png',
 'Non-TB/h1781.png',
 'Non-TB/h1791.png',
 'Non-TB/h1816.png',
 'Non-TB/h1826.png',
 'Non-TB/h1827.png',
 'Non-TB/h1832.png',
 'Non-TB/h1835.png',
 'Non-TB/h1843.png',
 'Non-TB/h1844.png',
 'Non-TB/h1864.png',
 'Non-TB/h1870.png',
 'Non-TB/h1874.png',
 'Non-TB/h1886.png',
 'Non-TB/h1919.png',
 'Non-TB/h1923.png',
 'Non-TB/h1933.png',
 'Non-TB/h1934.png',
 'Non-TB/h1950.png',
 'Non-TB/h1973.png',
 'Non-TB/h1975.png',
 'Non-TB/h1996.png',
 'Non-TB/h2000.png',
 'Non-TB/h2014.png',
 'Non-TB/h2020.png',
 'Non-TB/h2032.png',
 'Non-TB/h2043.png',
 'Non-TB/h2049.png',
 'Non-TB/h2066.png',
 'Non-TB/h2090.png',
 'Non-TB/h2098.png',
 'Non-TB/h2101.png',
 'Non-TB/h2108.png',
 'Non-TB/h2109.png',
 'Non-TB/h2120.png',
 'Non-TB/h2124.png',
 'Non-TB/h2138.png',
 'Non-TB/h2144.png',
 'Non-TB/h2148.png',
 'Non-TB/h2162.png',
 'Non-TB/h2164.png',
 'Non-TB/h2176.png',
 'Non-TB/h2178.png',
 'Non-TB/h2211.png',
 'Non-TB/h2282.png',
 'Non-TB/h2287.png',
 'Non-TB/h2294.png',
 'Non-TB/h2302.png',
 'Non-TB/h2309.png',
 'Non-TB/h2314.png',
 'Non-TB/h2320.png',
 'Non-TB/h2324.png',
 'Non-TB/h2327.png',
 'Non-TB/h2328.png',
 'Non-TB/h2332.png',
 'Non-TB/h2334.png',
 'Non-TB/h2350.png',
 'Non-TB/h2354.png',
 'Non-TB/h2361.png',
 'Non-TB/h2385.png',
 'Non-TB/h2407.png',
 'Non-TB/h2436.png',
 'Non-TB/h2451.png',
 'Non-TB/h2466.png',
 'Non-TB/h2467.png',
 'Non-TB/h2468.png',
 'Non-TB/h2472.png',
 'Non-TB/h2474.png',
 'Non-TB/h2479.png',
 'Non-TB/h2502.png',
 'Non-TB/h2525.png',
 'Non-TB/h2537.png',
 'Non-TB/h2540.png',
 'Non-TB/h2563.png',
 'Non-TB/h2592.png',
 'Non-TB/h2628.png',
 'Non-TB/h2633.png',
 'Non-TB/h2656.png',
 'Non-TB/h2667.png',
 'Non-TB/h2687.png',
 'Non-TB/h2698.png',
 'Non-TB/h2707.png',
 'Non-TB/h2710.png',
 'Non-TB/h2719.png',
 'Non-TB/h2738.png',
 'Non-TB/h2750.png',
 'Non-TB/h2766.png',
 'Non-TB/h2791.png',
 'Non-TB/h2800.png',
 'Non-TB/h2818.png',
 'Non-TB/h2830.png',
 'Non-TB/h2845.png',
 'Non-TB/h2866.png',
 'Non-TB/h2870.png',
 'Non-TB/h2872.png',
 'Non-TB/h2877.png',
 'Non-TB/h2884.png',
 'Non-TB/h2892.png',
 'Non-TB/h2894.png',
 'Non-TB/h2963.png',
 'Non-TB/h2985.png',
 'Non-TB/h2987.png',
 'Non-TB/h3005.png',
 'Non-TB/h3026.png',
 'Non-TB/h3045.png',
 'Non-TB/h3047.png',
 'Non-TB/h3055.png',
 'Non-TB/h3066.png',
 'Non-TB/h3080.png',
 'Non-TB/h3093.png',
 'Non-TB/h3109.png',
 'Non-TB/h3127.png',
 'Non-TB/h3144.png',
 'Non-TB/h3151.png',
 'Non-TB/h3152.png',
 'Non-TB/h3180.png',
 'Non-TB/h3192.png',
 'Non-TB/h3215.png',
 'Non-TB/h3254.png',
 'Non-TB/h3260.png',
 'Non-TB/h3277.png',
 'Non-TB/h3307.png',
 'Non-TB/h3313.png',
 'Non-TB/h3324.png',
 'Non-TB/h3351.png',
 'Non-TB/h3368.png',
 'Non-TB/h3399.png',
 'Non-TB/h3402.png',
 'Non-TB/h3410.png',
 'Non-TB/h3423.png',
 'Non-TB/h3428.png',
 'Non-TB/h3429.png',
 'Non-TB/h3477.png',
 'Non-TB/h3480.png',
 'Non-TB/h3486.png',
 'Non-TB/h3517.png',
 'Non-TB/h3521.png',
 'Non-TB/h3566.png',
 'Non-TB/h3572.png',
 'Non-TB/h3573.png',
 'Non-TB/h3576.png',
 'Non-TB/h3580.png',
 'Non-TB/h3581.png',
 'Non-TB/h3600.png',
 'Non-TB/h3609.png',
 'Non-TB/h3622.png',
 'Non-TB/h3637.png',
 'Non-TB/h3642.png',
 'Non-TB/h3707.png',
 'Non-TB/h3759.png',
 'Non-TB/h3763.png',
 'Non-TB/h3793.png',
 'Non-TB/h3797.png',
 'Non-TB/h3803.png',
 'Non-TB/h3805.png',
 'Non-TB/h3806.png',
 'Non-TB/h3822.png',
 'Non-TB/h3823.png',
 'Non-TB/h3837.png',
 'Non-TB/h3839.png',
 'Non-TB/h3851.png',
 'Non-TB/h3872.png',
 'Non-TB/h3875.png',
 'Non-TB/h3880.png',
 'Non-TB/h3896.png',
 'Non-TB/h3925.png',
 'Non-TB/h3932.png',
 'Non-TB/h3938.png',
 'Non-TB/h3962.png',
 'Non-TB/h4009.png',
 'Non-TB/h4012.png',
 'Non-TB/h4021.png',
 'Non-TB/h4035.png',
 'Non-TB/h4037.png',
 'Non-TB/h4042.png',
 'Non-TB/h4045.png',
 'Non-TB/h4046.png',
 'Non-TB/h4049.png',
 'Non-TB/h4066.png',
 'Non-TB/h4096.png',
 'Non-TB/h4108.png',
 'Non-TB/h4119.png',
 'Non-TB/h4120.png',
 'Non-TB/h4138.png',
 'Non-TB/h4142.png',
 'Non-TB/h4172.png',
 'Non-TB/h4189.png',
 'Non-TB/h4192.png',
 'Non-TB/h4212.png',
 'Non-TB/h4213.png',
 'Non-TB/h4250.png',
 'Non-TB/h4253.png',
 'Non-TB/h4261.png',
 'Non-TB/h4320.png',
 'Non-TB/h4327.png',
 'Non-TB/h4350.png',
 'Non-TB/h4369.png',
 'Non-TB/h4371.png',
 'Non-TB/h4383.png',
 'Non-TB/h4388.png',
 'Non-TB/h4406.png',
 'Non-TB/h4420.png',
 'Non-TB/h4426.png',
 'Non-TB/h4432.png',
 'Non-TB/h4439.png',
 'Non-TB/h4458.png',
 'Non-TB/h4461.png',
 'Non-TB/h4485.png',
 'Non-TB/h4489.png',
 'Non-TB/h4507.png',
 'Non-TB/h4519.png',
 'Non-TB/h4549.png',
 'Non-TB/h4571.png',
 'Non-TB/h4573.png',
 'Non-TB/h4575.png',
 'Non-TB/h4596.png',
 'Non-TB/h4598.png',
 'Non-TB/h4614.png',
 'Non-TB/h4665.png',
 'Non-TB/h4704.png',
 'Non-TB/h4707.png',
 'Non-TB/h4711.png',
 'Non-TB/h4717.png',
 'Non-TB/h4738.png',
 'Non-TB/h4786.png',
 'Non-TB/h4796.png',
 'Non-TB/h4855.png',
 'Non-TB/h4867.png',
 'Non-TB/h4894.png',
 'Non-TB/h4919.png',
 'Non-TB/h4920.png',
 'Non-TB/h4926.png',
 'Non-TB/h4938.png',
 'Non-TB/h4942.png',
 'Non-TB/h4944.png',
 'Non-TB/h4960.png',
 'Non-TB/h4962.png',
 'Non-TB/h4981.png',
 'Non-TB/h4983.png',
 'Non-TB/s0008.png',
 'Non-TB/s0021.png',
 'Non-TB/s0049.png',
 'Non-TB/s0054.png',
 'Non-TB/s0073.png',
 'Non-TB/s0101.png',
 'Non-TB/s0149.png',
 'Non-TB/s0164.png',
 'Non-TB/s0165.png',
 'Non-TB/s0178.png',
 'Non-TB/s0185.png',
 'Non-TB/s0200.png',
 'Non-TB/s0215.png',
 'Non-TB/s0238.png',
 'Non-TB/s0239.png',
 'Non-TB/s0245.png',
 'Non-TB/s0255.png',
 'Non-TB/s0265.png',
 'Non-TB/s0270.png',
 'Non-TB/s0274.png',
 'Non-TB/s0276.png',
 'Non-TB/s0288.png',
 'Non-TB/s0305.png',
 'Non-TB/s0316.png',
 'Non-TB/s0333.png',
 'Non-TB/s0336.png',
 'Non-TB/s0338.png',
 'Non-TB/s0370.png',
 'Non-TB/s0392.png',
 'Non-TB/s0398.png',
 'Non-TB/s0412.png',
 'Non-TB/s0416.png',
 'Non-TB/s0421.png',
 'Non-TB/s0430.png',
 'Non-TB/s0437.png',
 'Non-TB/s0445.png',
 'Non-TB/s0447.png',
 'Non-TB/s0457.png',
 'Non-TB/s0472.png',
 'Non-TB/s0498.png',
 'Non-TB/s0514.png',
 'Non-TB/s0517.png',
 'Non-TB/s0518.png',
 'Non-TB/s0537.png',
 'Non-TB/s0561.png',
 'Non-TB/s0582.png',
 'Non-TB/s0583.png',
 'Non-TB/s0603.png',
 'Non-TB/s0608.png',
 'Non-TB/s0623.png',
 'Non-TB/s0674.png',
 'Non-TB/s0681.png',
 'Non-TB/s0684.png',
 'Non-TB/s0690.png',
 'Non-TB/s0707.png',
 'Non-TB/s0709.png',
 'Non-TB/s0718.png',
 'Non-TB/s0725.png',
 'Non-TB/s0747.png',
 'Non-TB/s0758.png',
 'Non-TB/s0759.png',
 'Non-TB/s0764.png',
 'Non-TB/s0767.png',
 'Non-TB/s0806.png',
 'Non-TB/s0807.png',
 'Non-TB/s0815.png',
 'Non-TB/s0826.png',
 'Non-TB/s0870.png',
 'Non-TB/s0875.png',
 'Non-TB/s0885.png',
 'Non-TB/s0893.png',
 'Non-TB/s0895.png',
 'Non-TB/s0924.png',
 'Non-TB/s0961.png',
 'Non-TB/s0978.png',
 'Non-TB/s0990.png',
 'Non-TB/s1012.png',
 'Non-TB/s1016.png',
 'Non-TB/s1042.png',
 'Non-TB/s1045.png',
 'Non-TB/s1069.png',
 'Non-TB/s1072.png',
 'Non-TB/s1111.png',
 'Non-TB/s1144.png',
 'Non-TB/s1154.png',
 'Non-TB/s1155.png',
 'Non-TB/s1169.png',
 'Non-TB/s1175.png',
 'Non-TB/s1194.png',
 'Non-TB/s1219.png',
 'Non-TB/s1242.png',
 'Non-TB/s1251.png',
 'Non-TB/s1264.png',
 'Non-TB/s1276.png',
 'Non-TB/s1297.png',
 'Non-TB/s1309.png',
 'Non-TB/s1310.png',
 'Non-TB/s1318.png',
 'Non-TB/s1328.png',
 'Non-TB/s1346.png',
 'Non-TB/s1367.png',
 'Non-TB/s1371.png',
 'Non-TB/s1375.png',
 'Non-TB/s1384.png',
 'Non-TB/s1392.png',
 'Non-TB/s1409.png',
 'Non-TB/s1441.png',
 'Non-TB/s1447.png',
 'Non-TB/s1467.png',
 'Non-TB/s1486.png',
 'Non-TB/s1509.png',
 'Non-TB/s1526.png',
 'Non-TB/s1531.png',
 'Non-TB/s1538.png',
 'Non-TB/s1548.png',
 'Non-TB/s1555.png',
 'Non-TB/s1571.png',
 'Non-TB/s1587.png',
 'Non-TB/s1588.png',
 'Non-TB/s1597.png',
 'Non-TB/s1599.png',
 'Non-TB/s1611.png',
 'Non-TB/s1618.png',
 'Non-TB/s1630.png',
 'Non-TB/s1637.png',
 'Non-TB/s1640.png',
 'Non-TB/s1647.png',
 'Non-TB/s1666.png',
 'Non-TB/s1680.png',
 'Non-TB/s1693.png',
 'Non-TB/s1700.png',
 'Non-TB/s1702.png',
 'Non-TB/s1705.png',
 'Non-TB/s1708.png',
 'Non-TB/s1719.png',
 'Non-TB/s1728.png',
 'Non-TB/s1735.png',
 'Non-TB/s1751.png',
 'Non-TB/s1755.png',
 'Non-TB/s1765.png',
 'Non-TB/s1771.png',
 'Non-TB/s1792.png',
 'Non-TB/s1799.png',
 'Non-TB/s1802.png',
 'Non-TB/s1815.png',
 'Non-TB/s1819.png',
 'Non-TB/s1836.png',
 'Non-TB/s1846.png',
 'Non-TB/s1867.png',
 'Non-TB/s1873.png',
 'Non-TB/s1875.png',
 'Non-TB/s1880.png',
 'Non-TB/s1901.png',
 'Non-TB/s1916.png',
 'Non-TB/s1965.png',
 'Non-TB/s1969.png',
 'Non-TB/s1971.png',
 'Non-TB/s1977.png',
 'Non-TB/s1986.png',
 'Non-TB/s2017.png',
 'Non-TB/s2020.png',
 'Non-TB/s2024.png',
 'Non-TB/s2040.png',
 'Non-TB/s2052.png',
 'Non-TB/s2061.png',
 'Non-TB/s2085.png',
 'Non-TB/s2093.png',
 'Non-TB/s2138.png',
 'Non-TB/s2173.png',
 'Non-TB/s2177.png',
 'Non-TB/s2191.png',
 'Non-TB/s2194.png',
 'Non-TB/s2198.png',
 'Non-TB/s2200.png',
 'Non-TB/s2212.png',
 'Non-TB/s2214.png',
 'Non-TB/s2216.png',
 'Non-TB/s2228.png',
 'Non-TB/s2240.png',
 'Non-TB/s2255.png',
 'Non-TB/s2307.png',
 'Non-TB/s2311.png',
 'Non-TB/s2316.png',
 'Non-TB/s2317.png',
 'Non-TB/s2323.png',
 'Non-TB/s2326.png',
 'Non-TB/s2331.png',
 'Non-TB/s2344.png',
 'Non-TB/s2375.png',
 'Non-TB/s2384.png',
 'Non-TB/s2385.png',
 'Non-TB/s2395.png',
 'Non-TB/s2399.png',
 'Non-TB/s2440.png',
 'Non-TB/s2446.png',
 'Non-TB/s2448.png',
 'Non-TB/s2452.png',
 'Non-TB/s2456.png',
 'Non-TB/s2480.png',
 'Non-TB/s2505.png',
 'Non-TB/s2515.png',
 'Non-TB/s2516.png',
 'Non-TB/s2534.png',
 'Non-TB/s2547.png',
 'Non-TB/s2550.png',
 'Non-TB/s2557.png',
 'Non-TB/s2576.png',
 'Non-TB/s2593.png',
 'Non-TB/s2597.png',
 'Non-TB/s2635.png',
 'Non-TB/s2650.png',
 'Non-TB/s2665.png',
 'Non-TB/s2671.png',
 'Non-TB/s2681.png',
 'Non-TB/s2712.png',
 'Non-TB/s2739.png',
 'Non-TB/s2741.png',
 'Non-TB/s2742.png',
 'Non-TB/s2759.png',
 'Non-TB/s2771.png',
 'Non-TB/s2790.png',
 'Non-TB/s2794.png',
 'Non-TB/s2825.png',
 'Non-TB/s2853.png',
 'Non-TB/s2855.png',
 'Non-TB/s2872.png',
 'Non-TB/s2895.png',
 'Non-TB/s2907.png',
 'Non-TB/s2914.png',
 'Non-TB/s2919.png',
 'Non-TB/s2923.png',
 'Non-TB/s2929.png',
 'Non-TB/s2935.png',
 'Non-TB/s2947.png',
 'Non-TB/s3003.png',
 'Non-TB/s3024.png',
 'Non-TB/s3043.png',
 'Non-TB/s3050.png',
 'Non-TB/s3088.png',
 'Non-TB/s3096.png',
 'Non-TB/s3113.png',
 'Non-TB/s3134.png',
 'Non-TB/s3144.png',
 'Non-TB/s3153.png',
 'Non-TB/s3158.png',
 'Non-TB/s3160.png',
 'Non-TB/s3163.png',
 'Non-TB/s3165.png',
 'Non-TB/s3169.png',
 'Non-TB/s3173.png',
 'Non-TB/s3185.png',
 'Non-TB/s3199.png',
 'Non-TB/s3255.png',
 'Non-TB/s3260.png',
 'Non-TB/s3264.png',
 'Non-TB/s3278.png',
 'Non-TB/s3299.png',
 'Non-TB/s3313.png',
 'Non-TB/s3400.png',
 'Non-TB/s3421.png',
 'Non-TB/s3442.png',
 'Non-TB/s3443.png',
 'Non-TB/s3447.png',
 'Non-TB/s3451.png',
 'Non-TB/s3467.png',
 'Non-TB/s3524.png',
 'Non-TB/s3535.png',
 'Non-TB/s3546.png',
 'Non-TB/s3572.png',
 'Non-TB/s3587.png',
 'Non-TB/s3591.png',
 'Non-TB/s3601.png',
 'Non-TB/s3612.png',
 'Non-TB/s3628.png',
 'Non-TB/s3649.png',
 'Non-TB/s3651.png',
 'Non-TB/s3672.png',
 'Non-TB/s3681.png',
 'Non-TB/s3714.png',
 'Non-TB/s3728.png',
 'Non-TB/s3736.png',
 'Non-TB/s3738.png',
 'Non-TB/s3740.png',
 'Non-TB/s3744.png',
 'Non-TB/s3748.png',
 'Non-TB/s3752.png',
 'Non-TB/s3754.png',
 'Non-TB/s3756.png',
 'Non-TB/s3781.png',
 'Non-TB/s3784.png',
 'Non-TB/s3786.png',
 'Non-TB/s3788.png',
 'Non-TB/s3789.png',
 'Non-TB/s3807.png',
 'Non-TB/s3836.png',
 'Non-TB/s3847.png',
 'Non-TB/s3858.png',
 'Non-TB/s3865.png',
 'Non-TB/s3868.png',
 'Non-TB/s3890.png',
 'Non-TB/s3892.png',
 'Non-TB/s3910.png',
 'Non-TB/s3915.png',
 'Non-TB/s3921.png',
 'Non-TB/s3934.png',
 'Non-TB/s3935.png',
 'Non-TB/s3941.png',
 'Non-TB/s3946.png',
 'Non-TB/s3965.png',
 'Non-TB/s3970.png',
 'Non-TB/s3982.png',
 'Non-TB/s3991.png',
 'Non-TB/s4001.png',
 'Non-TB/s4009.png',
 'Non-TB/s4010.png',
 'Non-TB/s4016.png',
 'Non-TB/s4064.png',
 'Non-TB/s4069.png',
 'Non-TB/s4087.png',
 'Non-TB/s4144.png',
 'Non-TB/s4156.png',
 'Non-TB/s4157.png',
 'Non-TB/s4162.png',
 'Non-TB/s4193.png',
 'Non-TB/s4213.png',
 'Non-TB/s4214.png',
 'Non-TB/s4230.png',
 'Non-TB/s4231.png',
 'Non-TB/s4232.png',
 'Non-TB/s4241.png',
 'Non-TB/s4254.png',
 'Non-TB/s4270.png',
 'Non-TB/s4302.png',
 'Non-TB/s4309.png',
 'Non-TB/s4321.png',
 'Non-TB/s4327.png',
 'Non-TB/s4371.png',
 'Non-TB/s4384.png',
 'Non-TB/s4415.png',
 'Non-TB/s4416.png',
 'Non-TB/s4424.png',
 'Non-TB/s4447.png',
 'Non-TB/s4456.png',
 'Non-TB/s4458.png',
 'Non-TB/s4476.png',
 'Non-TB/s4483.png',
 'Non-TB/s4485.png',
 'Non-TB/s4491.png',
 'Non-TB/s4562.png',
 'Non-TB/s4563.png',
 'Non-TB/s4575.png',
 'Non-TB/s4590.png',
 'Non-TB/s4622.png',
 'Non-TB/s4634.png',
 'Non-TB/s4635.png',
 'Non-TB/s4640.png',
 'Non-TB/s4643.png',
 'Non-TB/s4648.png',
 'Non-TB/s4686.png',
 'Non-TB/s4689.png',
 'Non-TB/s4701.png',
 'Non-TB/s4716.png',
 'Non-TB/s4736.png',
 'Non-TB/s4737.png',
 'Non-TB/s4748.png',
 'Non-TB/s4764.png',
 'Non-TB/s4784.png',
 'Non-TB/s4839.png',
 'Non-TB/s4845.png',
 'Non-TB/s4904.png',
 'Non-TB/s4907.png',
 'Non-TB/s4911.png',
 'Non-TB/s4916.png',
 'Non-TB/s4919.png',
 'Non-TB/s4934.png',
 'Non-TB/s4961.png',
 'Non-TB/s4974.png',
 'Non-TB/s4987.png',
 'Non-TB/s4989.png',
 'TB/1.2.840.114062.2.192.168.196.13.2015.10.22.11.32.2.8847984.png',
 'TB/1.2.840.114062.2.192.168.196.13.2015.11.30.15.17.27.22945078.png',
 'TB/1.2.840.114062.2.192.168.196.13.2015.2.6.13.10.6.15062015.png',
 'TB/1.2.840.114062.2.192.168.196.13.2016.1.19.10.55.51.8840500.png',
 'TB/1.2.840.114062.2.192.168.196.13.2016.2.15.11.19.21.8648203.png',
 'TB/1.2.840.114062.2.192.168.196.14.2016.4.15.13.27.50.16957609.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.114596.20150303122009.12.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.2100.20150226021658.6.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.2100.20150226021724.44.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.2100.20150226021820.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.2100.20150226021831.39.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.2100.20150226021848.6.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.2100.20150226021859.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.2100.20150226021908.10.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.2100.20150226021908.13.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807020550.12.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807020626.4.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807020703.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807020751.6.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807020852.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021015.14.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021105.14.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021106.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021114.12.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021211.12.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021224.15.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021224.6.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021252.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021253.18.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021303.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021303.6.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021348.16.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021349.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021445.9.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021534.15.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021534.9.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021720.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021741.4.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807021941.8.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022012.17.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022012.8.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022013.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022022.13.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022114.12.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022114.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022115.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022115.9.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022232.5.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022233.9.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022234.12.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022329.6.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022534.21.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022534.6.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022547.6.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022548.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022620.12.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807022805.19.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807023020.6.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807023030.16.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807023122.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807023237.9.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807023452.6.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807023518.9.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807023600.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807023616.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807023955.5.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807024108.14.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807024159.16.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807024239.14.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807024239.6.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807024259.12.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807024323.10.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807024333.14.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807024411.16.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807024420.15.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807024439.8.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807024700.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807024847.18.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807025040.3.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.5060.20130807025250.9.png',
 'TB/1.3.6.1.4.1.25403.158515237678667.9208.20150310095211.3.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11044.20131119114251.100.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11044.20131119114329.179.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11044.20131119114405.3.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11044.20131119114516.16.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11044.20131119114529.17.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11044.20131119114529.8.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11044.20131119114533.65.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11044.20131119114545.26.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11044.20131119114551.53.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11044.20131119114552.3.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11044.20131119114603.36.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11044.20131119114619.74.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11044.20131119114619.77.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11044.20131119114627.15.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11044.20131119114629.3.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11044.20131119114636.6.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11492.20131121044930.10.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11492.20131121045002.16.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11492.20131121045002.19.png',
 'TB/1.3.6.1.4.1.25403.163683357445804.11492.20131121045002.31.png',
 'TB/1.3.6.1.4.1.25403.207732457674374.13668.20141127075841.92.png',
 'TB/1.3.6.1.4.1.25403.207732457674374.13668.20141127075841.95.png',
 'TB/1.3.6.1.4.1.25403.207732457674374.13668.20141127075940.24.png',
 'TB/1.3.6.1.4.1.25403.207732457674374.27968.20150929060020.95.png',
 'TB/1.3.6.1.4.1.25403.207732457674374.27968.20150929060037.55.png',
 'TB/1.3.6.1.4.1.25403.207732457674374.27968.20150929060047.44.png',
 'TB/1.3.6.1.4.1.25403.75984511895.34804.20150526025129.6.png',
 'TB/1.3.6.1.4.1.25403.75984511895.34804.20150526025132.102.png',
 'TB/1.3.6.1.4.1.25403.75984511895.34804.20150526025132.90.png',
 'TB/1.3.6.1.4.1.25403.75984511895.34804.20150526025139.46.png',
 'TB/1.3.6.1.4.1.25403.75984511895.34804.20150526025145.18.png',
 'TB/1.3.6.1.4.1.25403.75984511895.34804.20150526025152.33.png',
 'TB/1.3.6.1.4.1.25403.75984511895.34804.20150526025200.65.png',
 'TB/1.3.6.1.4.1.25403.75984511895.34804.20150526025219.9.png',
 'TB/CHNCXR_0356_1.png',
 'TB/CHNCXR_0358_1.png',
 'TB/CHNCXR_0363_1.png',
 'TB/CHNCXR_0364_1.png',
 'TB/CHNCXR_0365_1.png',
 'TB/CHNCXR_0366_1.png',
 'TB/CHNCXR_0372_1.png',
 'TB/CHNCXR_0373_1.png',
 'TB/CHNCXR_0376_1.png',
 'TB/CHNCXR_0385_1.png',
 'TB/CHNCXR_0406_1.png',
 'TB/CHNCXR_0418_1.png',
 'TB/CHNCXR_0419_1.png',
 'TB/CHNCXR_0420_1.png',
 'TB/CHNCXR_0465_1.png',
 'TB/CHNCXR_0480_1.png',
 'TB/CHNCXR_0482_1.png',
 'TB/CHNCXR_0502_1.png',
 'TB/CHNCXR_0539_1.png',
 'TB/CHNCXR_0556_1.png',
 'TB/CHNCXR_0568_1.png',
 'TB/CHNCXR_0572_1.png',
 'TB/CHNCXR_0607_1.png',
 'TB/CHNCXR_0608_1.png',
 'TB/CHNCXR_0633_1.png',
 'TB/CHNCXR_0658_1.png',
 'TB/MCUCXR_0108_1.png',
 'TB/MCUCXR_0141_1.png',
 'TB/MCUCXR_0162_1.png',
 'TB/MCUCXR_0251_1.png',
 'TB/p19.png',
 'TB/p22.png',
 'TB/pd446.png',
 'TB/pd846.png',
 'TB/pd861.png',
 'TB/px1.png',
 'TB/px17.png',
 'TB/px37.png',
 'TB/px49.png',
 'TB/px50.png',
 'TB/tb0020.png',
 'TB/tb0024.png',
 'TB/tb0027.png',
 'TB/tb0029.png',
 'TB/tb0037.png',
 'TB/tb0057.png',
 'TB/tb0089.png',
 'TB/tb0128.png',
 'TB/tb0146.png',
 'TB/tb0147.png',
 'TB/tb0156.png',
 'TB/tb0162.png',
 'TB/tb0170.png',
 'TB/tb0197.png',
 'TB/tb0201.png',
 'TB/tb0229.png',
 'TB/tb0282.png',
 'TB/tb0292.png',
 'TB/tb0293.png',
 'TB/tb0297.png',
 'TB/tb0299.png',
 'TB/tb0300.png',
 'TB/tb0327.png',
 'TB/tb0345.png',
 'TB/tb0364.png',
 'TB/tb0383.png',
 'TB/tb0418.png',
 'TB/tb0453.png',
 'TB/tb0456.png',
 'TB/tb0474.png',
 'TB/tb0485.png',
 'TB/tb0487.png',
 'TB/tb0514.png',
 'TB/tb0518.png',
 'TB/tb0535.png',
 'TB/tb0536.png',
 'TB/tb0558.png',
 'TB/tb0566.png',
 'TB/tb0592.png',
 'TB/tb0593.png',
 'TB/tb0606.png',
 'TB/tb0636.png',
 'TB/tb0663.png',
 'TB/tb0664.png',
 'TB/tb0666.png',
 …]
```

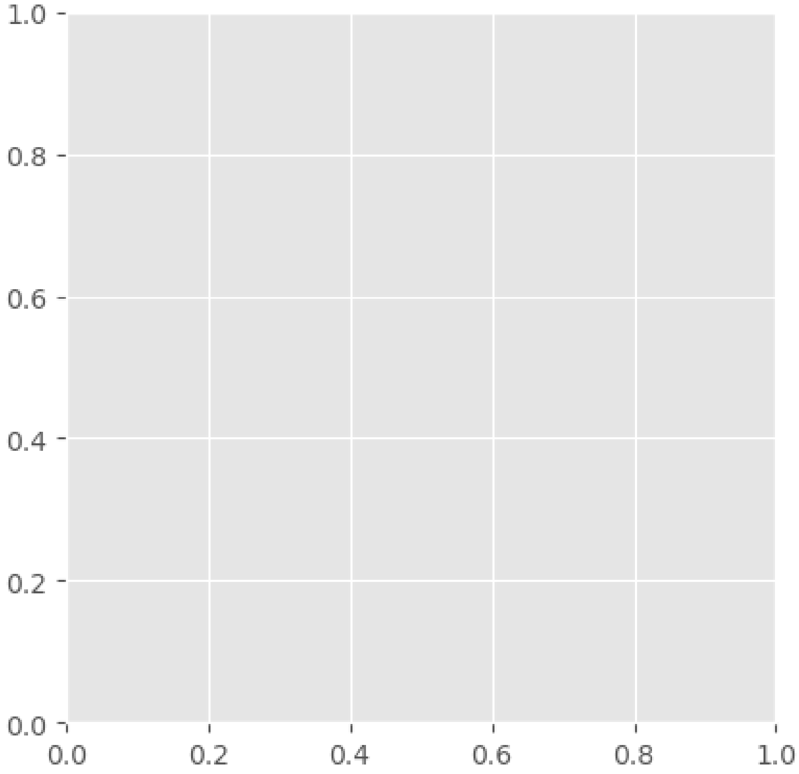
[ ]:

```
import glob
import cv2

# display sample of melanoma images
plt.style.use('ggplot')
paths = glob('../jpegs/split/train/1_mel/*.jpg')
fig, axes = plt.subplots(nrows=3, ncols=5, figsize=(16,6))
plt.suptitle('Melanoma Images', fontsize=16)
for i in range(0, 15):
    image = cv2.imread(paths[i], cv2.IMREAD_COLOR)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    image = cv2.resize(image, (256,256))
    x = i // 5
    y = i % 5
    axes[x, y].imshow(image, cmap=plt.cm.bone)
    axes[x, y].axis('off')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/Volumes/Steven 4TB M2 SSD/Dropbox/Omdena/
 ↪Tuberculosis-Image-Classification-Project/modeling/DenseNet121 - TB.ipynb Cell␣
 ↪18 line 4
      <a href='vscode-notebook-cell:/Volumes/Steven%204TB%20M2%20SSD/Dropbox/
 ↪Omdena/Tuberculosis-Image-Classification-Project/modeling/DenseNet121%20-%20TB.
 ↪ipynb#X23sZmlsZQ%3D%3D?line=1'>2</a> for i in range(4):
      <a href='vscode-notebook-cell:/Volumes/Steven%204TB%20M2%20SSD/Dropbox/
 ↪Omdena/Tuberculosis-Image-Classification-Project/modeling/DenseNet121%20-%20TB.
 ↪ipynb#X23sZmlsZQ%3D%3D?line=2'>3</a>     image = x[i]
```

```
----> <a href='vscode-notebook-cell:/Volumes/Steven%204TB%20M2%20SSD/Dropbox/
↪Omdena/Tuberculosis-Image-Classification-Project/modeling/DenseNet121%20-%20TB.
↪ipynb#X23sZmlsZQ%3D%3D?line=3'>4</a>     plt.imshow(image.astype('uint8'))
      <a href='vscode-notebook-cell:/Volumes/Steven%204TB%20M2%20SSD/Dropbox/
↪Omdena/Tuberculosis-Image-Classification-Project/modeling/DenseNet121%20-%20TB.
↪ipynb#X23sZmlsZQ%3D%3D?line=4'>5</a>     plt.show()

File ~/miniconda3/envs/tf-exam-env/lib/python3.9/site-packages/matplotlib/pyplot.
↪py:3346, in imshow(X, cmap, norm, aspect, interpolation, alpha, vmin, vmax,␣
↪origin, extent, interpolation_stage, filternorm, filterrad, resample, url,␣
↪data, **kwargs)
   3325 @_copy_docstring_and_deprecators(Axes.imshow)
   3326 def imshow(
   3327     X: ArrayLike | PIL.Image.Image,
   (…)
   3344     **kwargs,
   3345 ) -> AxesImage:
-> 3346     __ret = gca().imshow(
   3347         X,
   3348         cmap=cmap,
   3349         norm=norm,
   3350         aspect=aspect,
   3351         interpolation=interpolation,
   3352         alpha=alpha,
   3353         vmin=vmin,
   3354         vmax=vmax,
   3355         origin=origin,
   3356         extent=extent,
   3357         interpolation_stage=interpolation_stage,
   3358         filternorm=filternorm,
   3359         filterrad=filterrad,
   3360         resample=resample,
   3361         url=url,
   3362         **({"data": data} if data is not None else {}),
   3363         **kwargs,
   3364     )
   3365     sci(__ret)
   3366     return __ret

File ~/miniconda3/envs/tf-exam-env/lib/python3.9/site-packages/matplotlib/
↪__init__.py:1465, in _preprocess_data.<locals>.inner(ax, data, *args, **kwargs)
   1462 @functools.wraps(func)
   1463 def inner(ax, *args, data=None, **kwargs):
   1464     if data is None:
-> 1465         return func(ax, *map(sanitize_sequence, args), **kwargs)
   1467     bound = new_sig.bind(ax, *args, **kwargs)
   1468     auto_label = (bound.arguments.get(label_namer)
   1469                   or bound.kwargs.get(label_namer))
```

```
File ~/miniconda3/envs/tf-exam-env/lib/python3.9/site-packages/matplotlib/axes/
 ↪_axes.py:5751, in Axes.imshow(self, X, cmap, norm, aspect, interpolation,␣
 ↪alpha, vmin, vmax, origin, extent, interpolation_stage, filternorm, filterrad ␣
 ↪resample, url, **kwargs)
   5748 if aspect is not None:
   5749     self.set_aspect(aspect)
-> 5751 im.set_data(X)
   5752 im.set_alpha(alpha)
   5753 if im.get_clip_path() is None:
   5754     # image does not already have clipping set, clip to axes patch

File ~/miniconda3/envs/tf-exam-env/lib/python3.9/site-packages/matplotlib/image
 ↪py:723, in _ImageBase.set_data(self, A)
    721 if isinstance(A, PIL.Image.Image):
    722     A = pil_to_array(A)  # Needed e.g. to apply png palette.
--> 723 self._A = self._normalize_image_array(A)
    724 self._imcache = None
    725 self.stale = True

File ~/miniconda3/envs/tf-exam-env/lib/python3.9/site-packages/matplotlib/image
 ↪py:693, in _ImageBase._normalize_image_array(A)
    691     A = A.squeeze(-1)  # If just (M, N, 1), assume scalar and apply␣
 ↪colormap.
    692 if not (A.ndim == 2 or A.ndim == 3 and A.shape[-1] in [3, 4]):
--> 693     raise TypeError(f"Invalid shape {A.shape} for image data")
    694 if A.ndim == 3:
    695     # If the input data has values outside the valid range (after
    696     # normalisation), we issue a warning and then clip X to the bounds
    697     # - otherwise casting wraps extreme values, hiding outliers and
    698     # making reliable interpretation impossible.
    699     high = 255 if np.issubdtype(A.dtype, np.integer) else 1

TypeError: Invalid shape (16, 224, 224, 3) for image data
```



```
[ ]: pred
```

```
[ ]: array([[3.6021210e-02],
            [6.3000777e-04],
            [8.3144296e-05],
            …,
            [9.9936748e-01],
            [1.0000000e+00],
            [9.9897009e-01]], dtype=float32)
```

```
[ ]:
```