# -\*- coding: utf-8 -\*-

"""

Created on Thu Jan 13 05:29:45 2022

@author: WAEL

"""

import numpy as np

import pandas as pd

from pathlib import Path

import os.path

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

import tensorflow as tf

from sklearn.metrics import confusion\_matrix, classification\_report

image\_dir = Path('../content/drive/MyDrive/X-ray (2)/train')

filepaths = list(image\_dir.glob(r'\*\*/\*.png'))

labels = list(map(lambda x: os.path.split(os.path.split(x)[0])[1], filepaths))

filepaths = pd.Series(filepaths, name='Filepath').astype(str)

labels = pd.Series(labels, name='Label')

image\_df = pd.concat([filepaths, labels], axis=1)

train\_df, test\_df = train\_test\_split(image\_df, train\_size=0.7, shuffle=True, random\_state=1)

train\_generator = tf.keras.preprocessing.image.ImageDataGenerator(

    rescale=1./255,

    horizontal\_flip=True,

    width\_shift\_range=0.2,

    height\_shift\_range=0.2,

    validation\_split=0.2

)

test\_generator = tf.keras.preprocessing.image.ImageDataGenerator(

    rescale=1./255

)

train\_images = train\_generator.flow\_from\_dataframe(

    dataframe=train\_df,

    x\_col='Filepath',

    y\_col='Label',

    target\_size=(224, 224),

    color\_mode='rgb',

    class\_mode='binary',

    batch\_size=32,

    shuffle=True,

    seed=42,

    subset='training'

)

val\_images = train\_generator.flow\_from\_dataframe(

    dataframe=train\_df,

    x\_col='Filepath',

    y\_col='Label',

    target\_size=(224, 224),

    color\_mode='rgb',

    class\_mode='binary',

    batch\_size=32,

    shuffle=True,

    seed=42,

    subset='validation'

)

test\_images = test\_generator.flow\_from\_dataframe(

    dataframe=test\_df,

    x\_col='Filepath',

    y\_col='Label',

    target\_size=(224, 224),

    color\_mode='rgb',

    class\_mode='binary',

    batch\_size=32,

    shuffle=False

)

inputs = tf.keras.Input(shape=(224, 224, 3))

x = tf.keras.layers.Conv2D(filters=32, kernel\_size=(3, 3), activation='relu')(inputs)

x = tf.keras.layers.MaxPool2D()(x)

x = tf.keras.layers.Conv2D(filters=32, kernel\_size=(3, 3), activation='relu')(x)

x = tf.keras.layers.MaxPool2D()(x)

x = tf.keras.layers.Dropout(0.2)(x)

x =tf.keras.layers.Flatten()(x)

x = tf.keras.layers.Dense(32, activation='relu')(x)

x = tf.keras.layers.Flatten()(x)

outputs = tf.keras.layers.Dense(1, activation='sigmoid')(x)

model = tf.keras.Model(inputs=inputs, outputs=outputs)

model.compile(

    optimizer='adam',

    loss='binary\_crossentropy',

    metrics=['accuracy']

)

history = model.fit(train\_images,

    validation\_data=val\_images,

    epochs=100,

    callbacks=[

        tf.keras.callbacks.EarlyStopping(

            monitor='val\_loss',

            patience=5,

            restore\_best\_weights=True

        ),

        tf.keras.callbacks.ReduceLROnPlateau(

            monitor='val\_loss',

            patience=3

        )

    ]

)

results = model.evaluate(test\_images, verbose=0)

print("    Test Loss: {:.5f}".format(results[0]))

print("Test Accuracy: {:.2f}%".format(results[1] \* 100))

predictions = (model.predict(test\_images) >= 0.5).astype(np.int)

cm = confusion\_matrix(test\_images.labels, predictions, labels=[0, 1])

clr = classification\_report(test\_images.labels, predictions, labels=[0, 1], target\_names=["covid", "normal"])

plt.figure(figsize=(6, 6))

sns.heatmap(cm, annot=True, fmt='g', vmin=0, cmap='Blues', cbar=False)

plt.xticks(ticks=[0.5, 1.5], labels=["covid", "normal"])

plt.yticks(ticks=[0.5, 1.5], labels=["covid", "normal"])

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.title("Confusion Matrix")

plt.show()

print("Classification Report:\n----------------------\n", clr)

Found 316 validated image filenames belonging to 2 classes.

Found 78 validated image filenames belonging to 2 classes.

Found 169 validated image filenames belonging to 2 classes.

Epoch 1/100

10/10 [==============================] - 14s 1s/step - loss: 0.6779 - accuracy: 0.6677 - val\_loss: 0.6644 - val\_accuracy: 0.6667 - lr: 0.0010

Epoch 2/100

10/10 [==============================] - 12s 1s/step - loss: 0.5406 - accuracy: 0.7342 - val\_loss: 0.5663 - val\_accuracy: 0.6667 - lr: 0.0010

Epoch 3/100

10/10 [==============================] - 12s 1s/step - loss: 0.5099 - accuracy: 0.7658 - val\_loss: 0.5206 - val\_accuracy: 0.8205 - lr: 0.0010

Epoch 4/100

10/10 [==============================] - 12s 1s/step - loss: 0.5083 - accuracy: 0.7563 - val\_loss: 0.4869 - val\_accuracy: 0.7949 - lr: 0.0010

Epoch 5/100

10/10 [==============================] - 12s 1s/step - loss: 0.5429 - accuracy: 0.7089 - val\_loss: 0.6146 - val\_accuracy: 0.6795 - lr: 0.0010

Epoch 6/100

10/10 [==============================] - 12s 1s/step - loss: 0.5093 - accuracy: 0.7405 - val\_loss: 0.5336 - val\_accuracy: 0.6667 - lr: 0.0010

Epoch 7/100

10/10 [==============================] - 12s 1s/step - loss: 0.4773 - accuracy: 0.7437 - val\_loss: 0.5125 - val\_accuracy: 0.7692 - lr: 0.0010

Epoch 8/100

10/10 [==============================] - 12s 1s/step - loss: 0.4645 - accuracy: 0.7785 - val\_loss: 0.4491 - val\_accuracy: 0.7308 - lr: 1.0000e-04

Epoch 9/100

10/10 [==============================] - 12s 1s/step - loss: 0.4124 - accuracy: 0.7975 - val\_loss: 0.4363 - val\_accuracy: 0.7436 - lr: 1.0000e-04

Epoch 10/100

10/10 [==============================] - 12s 1s/step - loss: 0.4050 - accuracy: 0.8101 - val\_loss: 0.4763 - val\_accuracy: 0.7436 - lr: 1.0000e-04

Epoch 11/100

10/10 [==============================] - 12s 1s/step - loss: 0.3977 - accuracy: 0.8259 - val\_loss: 0.3896 - val\_accuracy: 0.7949 - lr: 1.0000e-04

Epoch 12/100

10/10 [==============================] - 12s 1s/step - loss: 0.3868 - accuracy: 0.8449 - val\_loss: 0.3596 - val\_accuracy: 0.8077 - lr: 1.0000e-04

Epoch 13/100

10/10 [==============================] - 12s 1s/step - loss: 0.3711 - accuracy: 0.8070 - val\_loss: 0.3952 - val\_accuracy: 0.8077 - lr: 1.0000e-04

Epoch 14/100

10/10 [==============================] - 12s 1s/step - loss: 0.3375 - accuracy: 0.8956 - val\_loss: 0.3698 - val\_accuracy: 0.8718 - lr: 1.0000e-04

Epoch 15/100

10/10 [==============================] - 12s 1s/step - loss: 0.3126 - accuracy: 0.8734 - val\_loss: 0.3110 - val\_accuracy: 0.8974 - lr: 1.0000e-04

Epoch 16/100

10/10 [==============================] - 12s 1s/step - loss: 0.3448 - accuracy: 0.8418 - val\_loss: 0.3433 - val\_accuracy: 0.8077 - lr: 1.0000e-04

Epoch 17/100

10/10 [==============================] - 12s 1s/step - loss: 0.3014 - accuracy: 0.8892 - val\_loss: 0.2948 - val\_accuracy: 0.8718 - lr: 1.0000e-04

Epoch 18/100

10/10 [==============================] - 12s 1s/step - loss: 0.3063 - accuracy: 0.8861 - val\_loss: 0.3129 - val\_accuracy: 0.8718 - lr: 1.0000e-04

Epoch 19/100

10/10 [==============================] - 12s 1s/step - loss: 0.2871 - accuracy: 0.9019 - val\_loss: 0.2634 - val\_accuracy: 0.9231 - lr: 1.0000e-04

Epoch 20/100

10/10 [==============================] - 12s 1s/step - loss: 0.2847 - accuracy: 0.8892 - val\_loss: 0.3155 - val\_accuracy: 0.9103 - lr: 1.0000e-04

Epoch 21/100

10/10 [==============================] - 12s 1s/step - loss: 0.2784 - accuracy: 0.8956 - val\_loss: 0.2652 - val\_accuracy: 0.9231 - lr: 1.0000e-04

Epoch 22/100

10/10 [==============================] - 12s 1s/step - loss: 0.2968 - accuracy: 0.8861 - val\_loss: 0.2886 - val\_accuracy: 0.8462 - lr: 1.0000e-04

Epoch 23/100

10/10 [==============================] - 12s 1s/step - loss: 0.2602 - accuracy: 0.8956 - val\_loss: 0.2416 - val\_accuracy: 0.9103 - lr: 1.0000e-05

Epoch 24/100

10/10 [==============================] - 12s 1s/step - loss: 0.2468 - accuracy: 0.9082 - val\_loss: 0.2752 - val\_accuracy: 0.9103 - lr: 1.0000e-05

Epoch 25/100

10/10 [==============================] - 12s 1s/step - loss: 0.2551 - accuracy: 0.9051 - val\_loss: 0.2770 - val\_accuracy: 0.9103 - lr: 1.0000e-05

Epoch 26/100

10/10 [==============================] - 12s 1s/step - loss: 0.2588 - accuracy: 0.9019 - val\_loss: 0.2567 - val\_accuracy: 0.9359 - lr: 1.0000e-05

Epoch 27/100

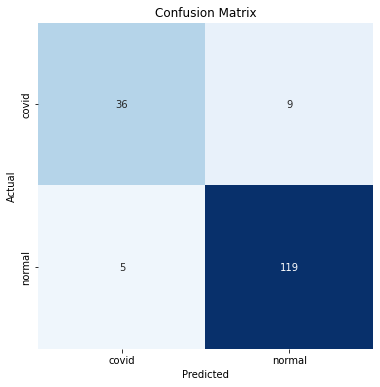
10/10 [==============================] - 12s 1s/step - loss: 0.2561 - accuracy: 0.9272 - val\_loss: 0.2499 - val\_accuracy: 0.9103 - lr: 1.0000e-06

Epoch 28/100

10/10 [==============================] - 12s 1s/step - loss: 0.2481 - accuracy: 0.9177 - val\_loss: 0.2729 - val\_accuracy: 0.9231 - lr: 1.0000e-06

Test Loss: 0.24291

Test Accuracy: 91.72%



Classification Report:

----------------------

precision recall f1-score support

covid 0.88 0.80 0.84 45

normal 0.93 0.96 0.94 124

accuracy 0.92 169

macro avg 0.90 0.88 0.89 169

weighted avg 0.92 0.92 0.92 169