**SOURCE CODES**

from os import path

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing.image import img\_to\_array, load\_img

import matplotlib.pyplot as plt

import numpy as np

import streamlit as st

@st.cache\_resource

def get\_trained\_model(model\_dir=None):

    model\_path = ''

    if model\_dir:

        model\_path = path.join(model\_dir, 'cataract\_classifier\_model.keras')

    else:

        model\_path = path.join('models', 'cataract\_classifier\_model.keras')

    return load\_model(model\_path)

def predict(model, img\_path, actual\_class='', show\_image=True):

    image = load\_img(img\_path, target\_size=(55,94))

    image = img\_to\_array(image)

    img = np.expand\_dims(image, axis=0)

    pred = model.predict(img, verbose=0)

    predicted\_class = 'normal' if pred[0] > 0.5 else 'cataract'

    results = ''

    if actual\_class:

        results = f'( Actual: {actual\_class}, Predicted: {predicted\_class} )'

    else:

        results = f'( Predicted: {predicted\_class} )'

    if show\_image:

        img = plt.imread(img\_path)

        plt.title(results)

        plt.imshow(img)

        plt.axis('off')

    else:

        print(results)

    return results

import streamlit as st

from os import path, listdir, remove

from utils import get\_trained\_model, predict

def clear\_dir(folder):

    files = listdir(folder)

    for file in files:

        file\_path = path.join(folder, file)

        if path.isfile(file\_path):

            # deletes file

            remove(file\_path)

UPLOAD\_FOLDER='uploads'

model = get\_trained\_model()

st.title('Cataract Detection using Transfer Learning Approach')

st.subheader('Fullname: Ojo Oluwaseun Emmanuel')

st.subheader('Matric Number: 2018/1/00021CS')

ct = st.container(border=True)

col1, col2 = ct.columns([0.6, 0.4])

ct1 = col1.container(border=True, height=300)

ct2 = col2.container(border=True, height=300)

uploaded\_image = ct1.file\_uploader('Upload the Image to Classify', type=['jpg','png'], accept\_multiple\_files=False)

btn\_classify = ct1.button('Classify Image', type='primary')

ct2.write('#### Classification Results')

if btn\_classify:

    clear\_dir(UPLOAD\_FOLDER)

    file\_path = path.join(UPLOAD\_FOLDER, uploaded\_image.name)

    with open(file\_path, 'wb') as f:

        f.write(uploaded\_image.getvalue())

    if path.exists(file\_path) and path.isfile(file\_path):

        predicted\_result = predict(model, file\_path, show\_image=False)

        ct2.write(predicted\_result)

        ct2.image(file\_path, use\_column\_width=True)

pred\_df.tail(20)

tp = tn = fp = fn = 0

for actual, predicted in zip(list(pred\_df['actual\_class']), list(pred\_df['predicted\_class'])):

if actual == 'normal' and predicted == 'normal':

tp = tp + 1

elif actual == 'cataract' and predicted == 'cataract':

tn = tn + 1

elif actual == 'normal' and predicted == 'cataract':

fp = fp + 1

elif actual == 'cataract' and predicted == 'normal':

fn = fn + 1

accuracy = (tp + tn) / (tp + tn + fp + fn)

precision = tp / (tp + fp)

recall = tp / (tp + fn)

f1 = 2 \* precision \* recall / (precision + recall)

evaluation\_summary = pd.DataFrame()

evaluation\_summary["accuracy"] = [accuracy]

evaluation\_summary["precision"] = [precision]

evaluation\_summary["recall"] = [recall]

evaluation\_summary["f1"] = [f1]

results = evaluation\_summary.T.round(2)

results.rename(columns={0:'scores'}, inplace=True)

results.sort\_values(by='scores', ascending=False).plot(kind='bar', title='Model Performance for Cataract Classifer Model\n', rot=0)

model.save("/kaggle/working/cataract\_classifier\_model.h5")

model.save("/kaggle/working/cataract\_classifier\_model.keras")

# # Loading trained models and Inferencing

import os

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing.image import img\_to\_array, load\_img

# from keras.preprocessing import image

import matplotlib.pyplot as plt

import numpy as np

def predict(model, img\_path, actual\_class='', show\_image=True):

image = load\_img(img\_path, target\_size=(55,94))

image = img\_to\_array(image)

img = np.expand\_dims(image, axis=0)

pred = model.predict(img, verbose=0)

predicted\_class = 'normal' if pred[0] > 0.5 else 'cataract'

results = ''

if actual\_class:

results = f'( Actual: {actual\_class}, Predicted: {predicted\_class} )'

else:

results = f'( Predicted: {predicted\_class} )'

if show\_image:

img = plt.imread(img\_path)

plt.title(results)

plt.imshow(img)

plt.axis('off')

else:

print(results)

model\_dir = '/kaggle/input/cataract-classifiers/keras/default/1'

trained\_model = load\_model(os.path.join(model\_dir, 'cataract\_classifier\_model.h5'))

testimage\_path = '/kaggle/input/cataract-image-dataset/processed\_images/test/cataract/image\_248.png'

predict(trained\_model, testimage\_path, actual\_class='cataract', show\_image=True)

testimage\_path = '/kaggle/input/cataract-image-dataset/processed\_images/test/normal/image\_250.png'

predict(trained\_model, testimage\_path, actual\_class='normal', show\_image=True)