import numpy as np

import tensorflow as tf

from tensorflow.keras.preprocessing import image

import matplotlib.pyplot as plt

from google.colab import files

import cv2

# Load Trained Model

model\_path = '/content/drive/MyDrive/melanoma\_bcc\_efficientnetv2\_algorithm.optimized.h5'

model = tf.keras.models.load\_model(model\_path)

# Class Labels

class\_labels = {0: 'Basal Cell Carcinoma', 1: 'Melanoma'}

# Function to Preprocess Image

def preprocess\_image(img\_path):

img = image.load\_img(img\_path, target\_size=(224, 224))

img\_array = image.img\_to\_array(img) / 255.0 # Normalize

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

return img\_array

# Function to Predict Image Class with "Unknown" Detection

def predict\_image(img\_path, threshold=0.7):

img\_array = preprocess\_image(img\_path)

prediction = model.predict(img\_array)[0]

# Handling Softmax Output

if len(prediction) == 2:

bcc\_prob, melanoma\_prob = prediction

max\_prob = max(bcc\_prob, melanoma\_prob)

else:

# Sigmoid Output Handling

max\_prob = max(prediction[0], 1 - prediction[0])

# If confidence is below threshold, classify as "Unknown"

if max\_prob < threshold:

return "Unknown / Not a Skin Cancer Lesion", max\_prob

# Otherwise, classify as Melanoma or BCC

predicted\_class = np.argmax(prediction) if len(prediction) == 2 else (1 if prediction[0] > 0.5 else 0)

return class\_labels[predicted\_class], max\_prob

# Function to Upload and Predict

def upload\_and\_predict():

uploaded = files.upload()

for filename in uploaded.keys():

img\_path = filename

predicted\_label, confidence = predict\_image(img\_path)

# Display Image & Prediction

img = cv2.imread(img\_path)

img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

plt.imshow(img)

plt.axis('off')

plt.title(f"Prediction: {predicted\_label} \n Confidence: {confidence:.2f}")

plt.show()

# Upload and Predict

upload\_and\_predict()