OBJECTIVE : Classify people with or without Real Time face masks in real settings.

CODE:

from tensorflow import keras

from tensorflow.keras import layers

from tensorflow.keras.preprocessing.image import ImageDataGenerator

import numpy as np

import cv2

IMG\_SIZE = 224

BATCH\_SIZE = 32

train\_datagen = ImageDataGenerator(rescale=1./255, validation\_split=0.2)

train\_generator = train\_datagen.flow\_from\_directory(

r'C:\Users\bdhan\Downloads\Face\_Mask\_Detection',

target\_size=(IMG\_SIZE, IMG\_SIZE),

batch\_size=BATCH\_SIZE,

class\_mode='binary',

subset='training'

)

val\_generator = train\_datagen.flow\_from\_directory(

r'C:\Users\bdhan\Downloads\Face\_Mask\_Detection',

target\_size=(IMG\_SIZE, IMG\_SIZE),

batch\_size=BATCH\_SIZE,

class\_mode='binary',

subset='validation'

)

print("Class indices:", train\_generator.class\_indices)

model = keras.Sequential([

layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(IMG\_SIZE, IMG\_SIZE, 3)),

layers.MaxPooling2D((2, 2)),

layers.Conv2D(64, (3, 3), activation='relu'),

layers.MaxPooling2D((2, 2)),

layers.Conv2D(128, (3, 3), activation='relu'),

layers.MaxPooling2D((2, 2)),

layers.Flatten(),

layers.Dense(128, activation='relu'),

layers.Dense(1, activation='sigmoid')

])

model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

model.summary()

model.fit(train\_generator, epochs=5, validation\_data=val\_generator)

model.save('Real\_World\_Mask\_Detection.h5')

model = keras.models.load\_model('Real\_World\_Mask\_Detection.h5')

print("Model Loaded")

face\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + "haarcascade\_frontalface\_default.xml")

cap = cv2.VideoCapture(0)

while True:

ret, frame = cap.read()

if not ret:

break

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

faces = face\_cascade.detectMultiScale(gray, 1.1, 4)

for (x, y, w, h) in faces:

face = frame[y:y+h, x:x+w]

try:

face\_resized = cv2.resize(face, (IMG\_SIZE, IMG\_SIZE))

face\_array = face\_resized / 255.0

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

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

label = "No Mask" if prediction < 0.5 else "Mask"

color = (0, 255, 0) if label == "Mask" else (0, 0, 255)

cv2.rectangle(frame, (x, y), (x+w, y+h), color, 2)

cv2.putText(frame, label, (x, y-10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.8, color, 2)

except Exception as e:

print("Error processing face:", e)

continue

cv2.imshow("Face Mask Detector", frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release()

cv2.destroyAllWindows()

OUTPUT:



