Face Emotion Detection

pip install opencv-python

pip install face\_recognition

pip install imutils

pip install deepface

from deepface import DeepFace

import numpy as np

import cv2

import imutils

from IPython.display import display, Javascript

from google.colab.output import eval\_js

from base64 import b64decode

from google.colab.patches import cv2\_imshow

import ipywidgets as widgets

from IPython.display import display

from PIL import Image

import io

import cv2

from google.colab.patches import cv2\_imshow

import numpy as np

from IPython.display import display,Markdown

#create a header and sub header

project\_name=Markdown("#Face Emotion Detection")

subheader=Markdown("###Click an image to identify the emotion!")

#display ui elements

display(project\_name)

display(subheader)

#create button widget

button = widgets.Button(description='Click here',button\_style="info")

output = widgets.Output()

# Define event handlers

def on\_button\_click(button):

button.description = "Clicked" if button.description=="Click here" else "Clicked"

button.button\_style="info" if button.description=="Click here" else "success"

predict()

#button.on\_click(on\_button\_click)

# Display UI elements

display(button)

display(output)

def predict():

from IPython.display import display, Javascript

from google.colab.output import eval\_js

from base64 import b64decode

def take\_photo(filename='photo.jpg', quality=0.8):

js = Javascript('''

async function takePhoto(quality) {

const div = document.createElement('div');

const capture = document.createElement('button');

capture.textContent = 'Capture';

div.appendChild(capture);

const video = document.createElement('video');

video.style.display = 'block';

const stream = await navigator.mediaDevices.getUserMedia({video: true});

document.body.appendChild(div);

div.appendChild(video);

video.srcObject = stream;

await video.play();

// Resize the output to fit the video element.

google.colab.output.setIframeHeight(document.documentElement.scrollHeight, true);

// Wait for Capture to be clicked.

await new Promise((resolve) => capture.onclick = resolve);

const canvas = document.createElement('canvas');

canvas.width = video.videoWidth;

canvas.height = video.videoHeight;

canvas.getContext('2d').drawImage(video, 0, 0);

stream.getVideoTracks()[0].stop();

div.remove();

return canvas.toDataURL('image/jpeg', quality);

}

''')

display(js)

data = eval\_js('takePhoto({})'.format(quality))

binary = b64decode(data.split(',')[1])

with open(filename, 'wb') as f:

f.write(binary)

return filename

from IPython.display import Image

try:

filename = take\_photo()

print('Saved to {}'.format(filename))

# Show the image which was just taken.

display(Image(filename))

except Exception as err:

# Errors will be thrown if the user does not have a webcam or if they do not

# grant the page permission to access it.

print(str(err))

import cv2

import matplotlib.pyplot as plt

img = cv2.imread("/content/photo.jpg")

#Analyze the face

predictions = DeepFace.analyze(img, actions = ['emotion'])

print(predictions)

#Display the dominant emotion and gender on image window and show the image

p = predictions[0]['dominant\_emotion']

i = cv2.putText(img, p, (10,40), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (255,255,255), 2, cv2.LINE\_AA)

cv2\_imshow(img)

button.on\_click(on\_button\_click)