**Face Mood Recognition**

***- Deepankar Sharma***

**[Face Emotion Recognizer](https://github.com/justinshenk/fer)**

detect\_emotions() -----> {‘fear’, ‘neutral’, ‘happy’, ’sad’, ‘anger’, ‘disgust’}.

Every emotion is calculated, and the output is put on a scale of 0 to 1.

import cv2

import streamlit as st

import os

import numpy as np

from keras.preprocessing import image

import warnings

warnings.filterwarnings("ignore")

from keras.utils import load\_img, img\_to\_array

from keras.preprocessing import image

from keras.models import  load\_model

import matplotlib.pyplot as plt

from fer import FER

emotion\_detector= FER()

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

st.title("Face Emotion Recognition System")

col1, col2= st.columns((5,5))

with col1:

    start= st.button("Start live input feed")

with col2:

    stop= st.button("Stop live input feed")

FRAME\_WINDOW = st.image([])

if start:

    camera = cv2.VideoCapture(0)

    while True:

        \_, frame = camera.read()

        rgb\_img = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

        faces\_detected = face\_haar\_cascade.detectMultiScale(rgb\_img, 1.32, 5)

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

            cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), *thickness*=7)

            roi\_rgb = rgb\_img[y:y + w, x:x + h]  # cropping region of interest i.e. face area from  image

            roi\_rgb = cv2.resize(roi\_rgb, (224, 224))

            img\_pixels = img\_to\_array(roi\_rgb)

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

            img\_pixels /= 255

            predictions= emotion\_detector.detect\_emotions(rgb\_img)

            predicted\_emotion, score= emotion\_detector.top\_emotion(rgb\_img)

            emotion\_score = "You look {}: ({})".format(predicted\_emotion, "{*:.2f*}".format(score))

            cv2.putText(frame, emotion\_score, (int(x), int(y)), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 0, 255), 2)

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

        FRAME\_WINDOW.image(frame)

        if stop:

            break