DETECT FACES AND ANALYSE EMOTIONS USING FACIAL EMOTION RECOGNITION API WITH IBM CLOUD

INDEX:

1.INTRODUCTION:

a) Overview:

We know that emotions play a major role in a Human life. At different kind of moments or time Human face reflects that how he/she feels or in which mood he/she is. Humans are capable of producing thousands of facial actions during communication that vary in complexity, intensity, and meaning. Emotion or intention is often communicated by subtle

changes in one or several discrete features. The addition or absence of one or more facial actions may alter its interpretation. In addition, some facial expressions may have a similar gross morphology but indicate varied meaning for different expression intensities. In order to capture the subtlety of facial expression in non-verbal communication, I will use an existing simulator which will be able to capture human emotions by reading or comparing facial expressions. This algorithm automatically extracts features and their motion information, discriminate subtly different facial expressions, and estimate expression intensity.

b) Purpose:

The main aim of this project is to detect face and should analyse the emotion of that face. This project will be very useful in the real time environment. By using Api's we can detect the emotions. Human emotion recognition plays an important role in the interpersonal relationship. Emotions are reflected from speech, hand and gestures of the body and through facial expressions. Hence extracting and understanding of emotion has a high importance of the interaction between human and machine communication.

2.LITERATURE SURVEY:

a) Existing problem:

Sometimes we should analyze emotions from image. But we are having less resources to analyse emotions from the image. Because of less resources it is difficult to analyse emotion from image. We humans will have mixed emotions and emotions

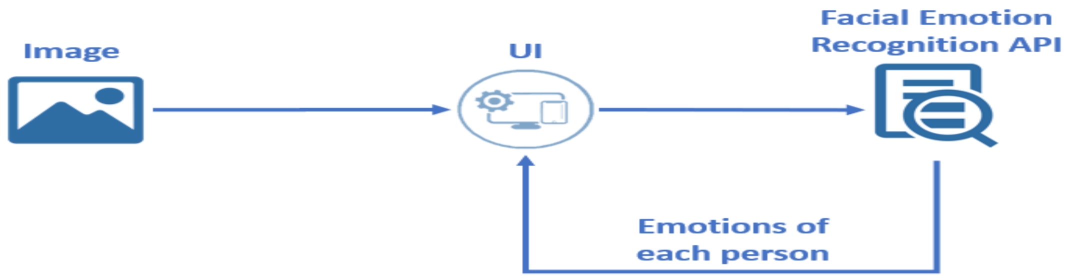
change time to time due to this it is very difficult to analyse emotions. If a person wants to know to emotion of a particular person in the photo then user must search for the different softwares to analyse emotions and some softwares are difficult to understand and some are time consuming process.Though humans have mixed emotions it is difficult to understand the emotions.There are many softwares are applications which are difficult to understand and it takes more time so to resolve this we have developed the project which is detect faces and analyse emotions using facial recognition API.

b) Proposed solution:

Human facial expressions can be easily classified into 7 basic emotions: happy, sad, surprise, fear, anger, blurred, and under exposed. Our facial emotions are expressed through activation of specific sets of facial muscles. These sometimes subtle, yet complex, signals in an expression often contain an abundant amount of information about our state of mind. Through facial emotion recognition, we are able to measure the effects that content and services have on the audience/users through an easy and low-cost procedure. For example, retailers may use these metrics to evaluate customer interest. Healthcare providers can provide better service by using additional information about patients' emotional state during treatment. Entertainment producers can monitor audience engagement in events to consistently create desired content. Humans are well-trained in reading the emotions of others, in fact, at just 14 months old, babies can already tell the difference between happy and sad. But can computers do a better job than us in accessing emotional states? To answer the question, We designed a detect faces and analyse emotions using facial emotion recognition API that gives machines the ability to make inferences about our emotional states. In other words, we give them eyes to see what we can see.

3.THEOTRICAL ANALYSIS:

a) Block diagram:



b) Hardware/Software designing

Software Requirements:

- OS-Windows XP,7,8,10
- Spyder Software
- Anaconda Command Prompt
- Flask framework
- Browser

Hardware Components

- Processor – i3
- Hard Disk Storage – 5GB
- RAM – 1GB

Required packages:

1. Open anaconda prompt.
2. Type "pip install requests” and click enter.
3. Type “pip install Flask” and click enter.

4.EXPERIMENTAL INVESTIGATIONS:

Experimental Investigations refers to research papers which helped us to implement with the help of flask framework as well as usage of rapid API. And also using IBM cloud which is a good platform which helps to deploy models.

IBM Watson on the IBM Cloud helps to transform businesses, enhancing competitive advantage and disrupting industries by unlocking the potential within unstructured data. Fundamental to providing a strong foundation for companies wanting to leverage Watson

IBM uses best-in-class security and compliance processes that allow for successful execution of challenging workloads. IBM Cloud Deployment Services (ICDS) reduces complexity by integrating multi cloud environments with a single orchestration platform.

Flask: Web framework used for building Web applications.

API: An application programming interface (API) is a computing interface which defines interactions between multiple software intermediaries.

4. WORK FLOW:

- User interacts with the UI(User Interface) to upload image Url as input

- The input is processed by measuring the dimension in the face in the given image with the help of an emotion recognition API.

- Then by calculating the dimensions in the face it gives emotion as output.

6.RESULT:

Gives output a emotion by using facial emotion recognition API.

7.ADVANTAGES AND DISADVANTAGES:

Advantages:

- Automatically detects the emotion.
- At a time it can detect one or more persons face and can analyse emotions.
- It is easy to use.

Disadvantages:

- It does not give accurate results all the time.

- It does not analyse the emotion of blurred image

8.Applications:

- ❖ Detection of facial emotions.
- ❖ Synthetic human expressions

9.CONCLUSION:

- By doing this project which is detect faces and and analyse emotion recognition API with IBM cloud, we have learnt how to use APIs, how to hit the APIs, send and retrieve the data and how to build web applications using flask framework. We also learnt how API works. We also learnt how to build applications using flask framework.
- We learnt how to use APIs and learnt about the flask framework.

10.FUTURE SCOPE:

In future we can develop this project more and we can increase the accuracy in analysing the emotions.This project can be used in many fields if we increase accuracy.it will be more useful in the present society.

11.BIBILOGRAPHY:

· Bassili J. N. (1979). Emotion recognition: The role of facial movement and the relative importance of upper and lower areas of the face. Journal of Personality and Social Psychology, 37, 2049–2058. [CrossRef] [PubMed]

· Berry D. S. (1990). What can a moving face tell us? Journal of Personality and SocialPsychology, 58, 1004–1014. [CrossRef] [PubMed]

· Blair R. J. R. Colledge E. Murray L. Mitchell D. G. V. (2001). A selective impairment in the processing of sad and fearful expressions in children with psychopathic tendencies. Journal of Abnormal Child Psychology, 29, 491–498. [CrossRef] [PubMed]

12.SOURCE CODE(APP.PY)

```
from flask import Flask, request, render_template

import numpy as np

import re

import requests

import pandas as pd
```

```
#initializing the flask app

app = Flask(__name__)


def check(output):


    url = "https://facial-emotion-recognition.p.rapidapi.com/cloudVision/facialEmotionRecognition"


    querystring = {"source":output,"sourceType":"url"}


    payload = '''{\r\n    \"source\": "'''+output+'''" ,\r\n    \"sourceType\": \"url\"\r\n}'''

    headers = {

    'content-type': "application/json",

    'x-rapidapi-key': "d730d682demshbfaf2fc584babdap15184djsnd2eec3eec4be",

    'x-rapidapi-host': "facial-emotion-recognition.p.rapidapi.com"

    }

    response = requests.request("POST", url, data=payload, headers=headers, params=querystring)

    print(response.text)

    a = response.json()['emotions']

    #converting the output to a dataframe using pandas

    df = pd.DataFrame(data=a)
```

```
        b = [i for i in range(1,len(a)+1)]
        ind={}
        for i in b:
        ind[i-1]="Person {:d}".format(i)
        #renaming the index and column names
        df=df.rename(index=ind)
        df.columns = ['Joy','Sorrow','Anger','Surprise','Under Exposed','Blurred','Hear Wear']
  df=df.replace(['VERY_UNLIKELY','UNLIKELY','VERY_LIKELY','LIKELY'], ["Very Unlikely","Unlikely","Very Likely","Likely"])
        #converting the dataframe to HTML table
        df_html = df.to_html(classes='mystyle', justify="center",col_space=90)

        return df_html



#home page
@app.route('/')
def home():
        return render_template('base.html')

#Summarizer page
@app.route('/predict',methods=['POST'])
```

```
def predict():

    output=request.form['output']

    df_html=check(output)

    return render_template('base.html',table_html=df_html)




if __name__ == "__main__":

    app.run(debug=True)
```

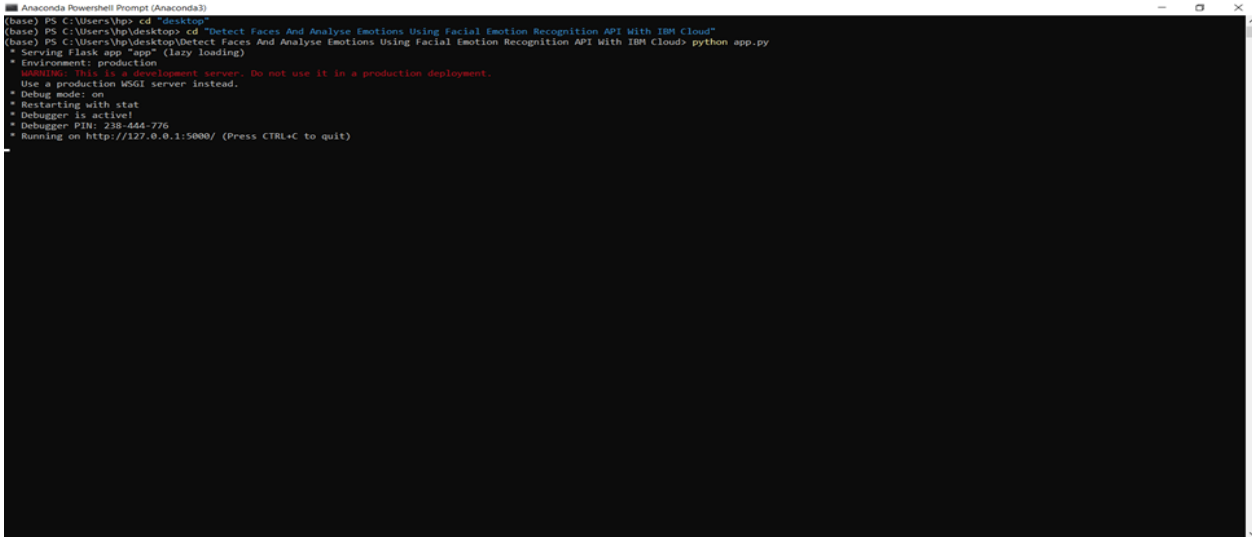
SOURCE CODE(EMOTION.PY):

```
import requests

url ="https://facial-emotion-recognition.p.rapidapi.com/cloudVision/facialEmotionRecognition"



querystring = {"source":"https://images.unsplash.com/photo-1527631120902-378417754324?ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=2250&q=80","sourceType":"url"}



payload = "{\r\n    \"source\": \"https://images.unsplash.com/photo-1527631120902-378417754324?ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=2250&q=80\",\r\n \"sourceType\": \"url\"\r\n}"

headers = {

    'content-type': "application/json",
'x-rapidapi-key': "8671a96a61msh86565f63e6a7c81p1911d5jsnc641b6e93b35",

'x-rapidapi-host': "facial-emotion-recognition.p.rapidapi.com"

}
```

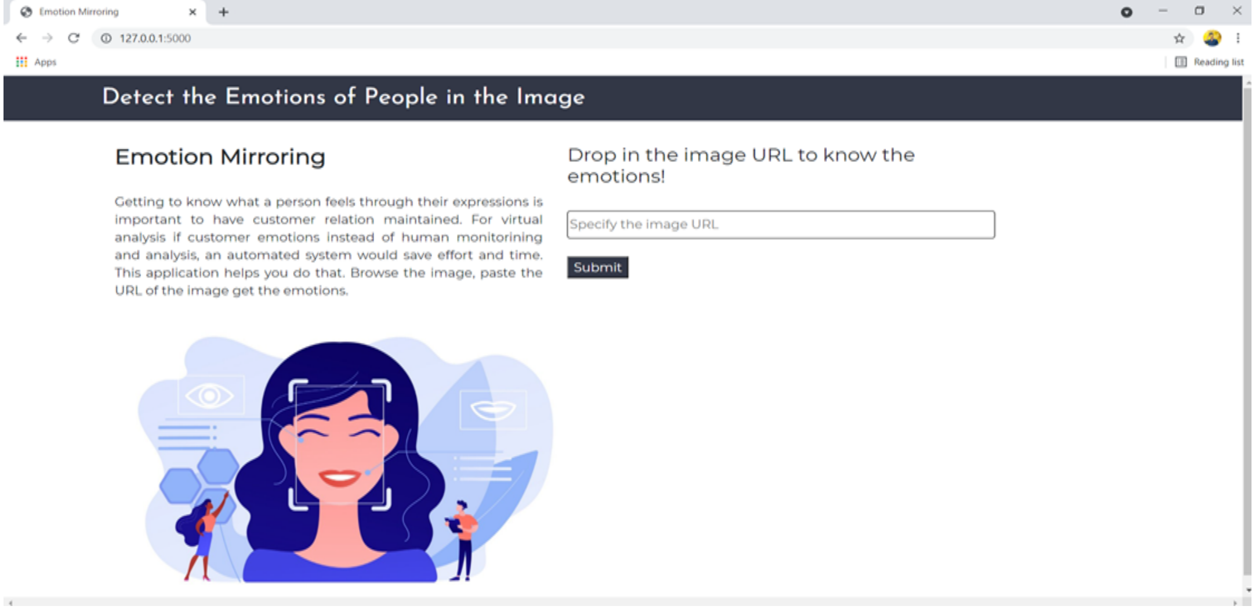
```
response = requests.request("POST", url, data=payload, headers=headers,
params=querystring)
```

```
print(response.text)
```

UI Output Screenshot:



```
(base) PS C:\Users\hp> cd "desktop"
(base) PS C:\Users\hp\desktop> cd "Detect Faces And Analyse Emotions Using Facial Emotion Recognition API With IBM Cloud"
(base) PS C:\Users\hp\desktop\Detect Faces And Analyse Emotions Using Facial Emotion Recognition API With IBM Cloud> python app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 238-444-776
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```



Detect the Emotions of People in the Image

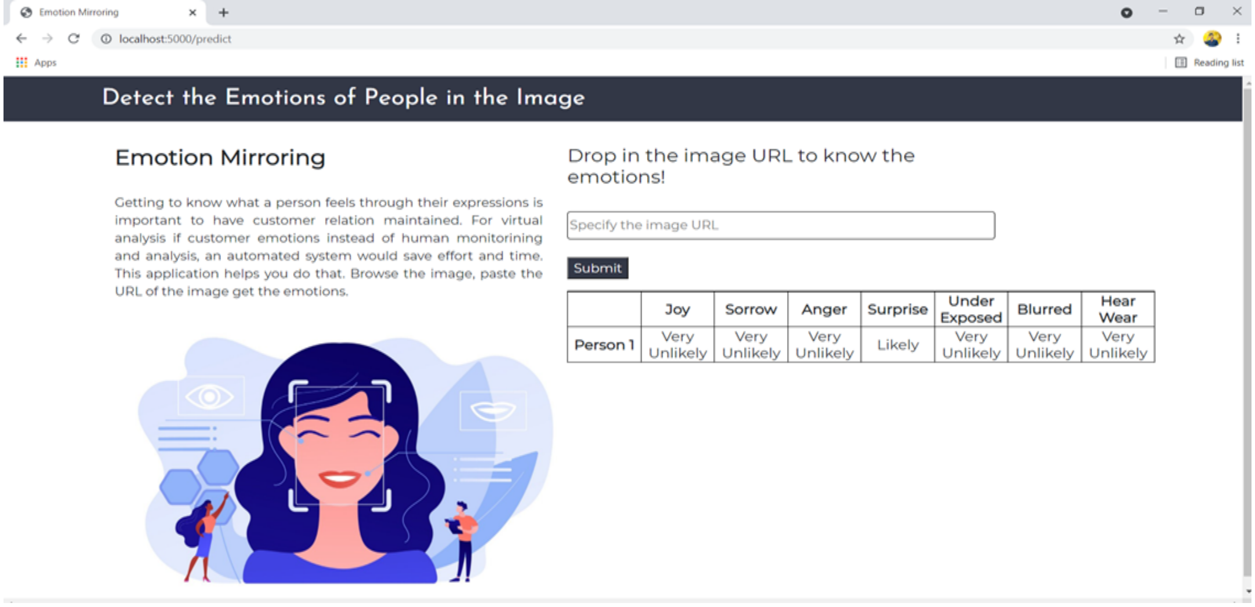
Emotion Mirroring

Getting to know what a person feels through their expressions is important to have customer relation maintained. For virtual analysis if customer emotions instead of human monitoring and analysis, an automated system would save effort and time. This application helps you do that. Browse the image, paste the URL of the image get the emotions.

Drop in the image URL to know the emotions!

Specify the image URL

Submit



Detect the Emotions of People in the Image

Emotion Mirroring

Getting to know what a person feels through their expressions is important to have customer relation maintained. For virtual analysis if customer emotions instead of human monitoring and analysis, an automated system would save effort and time. This application helps you do that. Browse the image, paste the URL of the image get the emotions.

Drop in the image URL to know the emotions!

Specify the image URL

Submit

	Joy	Sorrow	Anger	Surprise	Under Exposed	Blurred	Hear Wear
Person 1	Very Unlikely	Very Unlikely	Very Unlikely	Likely	Very Unlikely	Very Unlikely	Very Unlikely