

**AI in Enterprise System**  
**Facial emotion detection for customer relationship management**  
**Final Document**

**Maviya Javed Shaikh - 100766785**  
**Meryl Gabrielle Tubio – 100763231**  
**Nandini Malhotra – 100768797**

**April 8, 2021**

## Contents

I.	Problem definition and Project Pitch .....	3
II.	Exploratory Data Analysis .....	3
a.	Facial Expression Recognition .....	3
b.	Methodology .....	6
c.	Convolution Neural Network.....	7
d.	Data .....	13
III.	Solution Development .....	13
a.	Back-end and Front-end.....	13
b.	Software and Hardware Requirements .....	14
c.	Deploying it on docker.....	14
IV.	Project Timelines .....	20
V.	References .....	20

**No table of figures entries found.**

## **I. Problem definition and Project Pitch**

Facial expression recognition is a relatively recent technology in computer vision that is built around the idea of examining facial nuances of an individual's emotional reaction. Moreover, facial recognition Facial expression recognition is an essential ability for good interpersonal relations (Niedenthal and Brauer, 2012), and a major subject of study in the fields of human development, psychological well-being, and social adjustment. In fact, emotion recognition plays a pivotal role in the experience of empathy (Gery et al., 2009), in the prediction of prosocial behavior (Marsh et al., 2007), and in the ability model of emotional intelligence (Salovey and Mayer, 1990).

The task is to categorize each face based on the emotion shown in the facial expression in to one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).

## **II. Exploratory Data Analysis**

### **a. Facial Expression Recognition**

Identifying facial expressions is a highly effective form of catching non-verbal communication. Studies show that new mothers are the most sensitive to changes in another person's facial expression and body language. This is evidence that facial expressions are closely linked to emotional state, mindset, and intention.

Recognizing minute changes in facial muscles can vastly alter the flow of conversation by providing real-time feedback to speakers on how the audience is consuming their words. This is also beneficial for psychological research and in the gaming industry. As this allows subtle cues to be captured and further studied by the researchers.

One simple example of facial expression application is when a test subject creates a mismatch of the facial expression and the spoken word. The information passed along will gain more importance since this includes the

Facial expressions not only change the flow of conversation but also provides the listeners a way to communicate a wealth of information to the speaker without even uttering a single word. When the facial expression does not match with the spoken words, then the information pass on by the face gets more power in interpreting the information. From the perspective of automatic recognition, a facial expression can be considered to consist of deformations of facial components and their spatial relations, or changes in the pigmentation of the face. Facial expressions represent the changes of facial appearance in reaction to a person's inside emotional states, social communications or intentions.

To communicate the emotions and express the intentions the Facial expression is the most powerful, natural, non verbal and instant way for humans. It is faster to communicate the emotions through facial expressions than through verbalization. The requirement for proficient communication channels between machines and humans becomes progressively imperative in light of the fact that machines and individuals start to share a variety of tasks. Systems to form these communication channels are known as human machine interaction (HMI) systems. Progresses in technology make it possible the development of more useful HMI systems which no more depend on regular devices for example keyboard, mouse and displays but take commands directly from user's voice and mimics. Such systems intend to simulate human-human interaction by only utilizing communication channels utilized between humans and not requiring artificial equipment.

Human-machine interaction should be enhanced to more nearly simulate human to human interaction before machines take more places in our lives. As change of expressions on human face is an intense method for passing emotions, facial expression recognition (FER) will be one of the best steps for improving HMI systems. An automatic facial expression recognition system generally comprises of three main parts: face detection, facial feature points extraction and facial expression classification. In the first step, system obtains input image and performs some image processing techniques on it in order to locate the face region. In static images it is called face localization whereas in videos it is called face tracking. After the face has been located in the image or video frame, it can be evaluated in terms of facial action happening.

A feature is a point of interest or a piece of information. There are two types of features that are generally used to represent facial expression: geometric features and appearance features. Geometric features evaluate the displacements of certain parts of the face for instance brows or mouth corners, whereas appearance features represent the change in texture of face when particular action is performed. The task of geometric feature measurement is typically associated with face region analysis, particularly finding, and tracking key points in the face region. Potential problems that take place in face decomposition task could be occlusions and occurrences of facial hair or glasses. Moreover, defining the feature set is difficult, because features should be descriptive and possibly not correlated. The last part of the Facial Expressions Recognition system is based on machine learning theory; specifically, it is the classification task. A set of features which were retrieved from face region in the previous stage is the input to the classifier. The set of features is created to explain the facial expression.

Classification needs supervised training, so the training set should consist of labeled data. Once the classifier is trained, it can recognize input images by assigning them a specific class label. The most frequently used facial expressions classification is done both in terms of Action Units, proposed in Facial Action Coding System and in terms of universal emotions: happiness, sadness, anger, surprise, disgust, and fear. There are several different machine learning techniques for classification task for instance K-Nearest Neighbors, Artificial Neural Networks, Support Vector Machines, Hidden Markov Models, Expert Systems with rule-based classifier, Bayesian Networks or Boosting Techniques. Three main issues in classification task are: choosing good feature set, competent machine learning technique and different database for training. Feature set should be

composed of features that are discriminative and characteristic for specific expression. Machine learning technique is selected usually by the type of a feature set.

The database used as a training set should be big enough and include a variety of data. In facial expression recognition Region of Interests represent the eye pair, nostrils, and the mouth area. Region of Interest is related to define a large region which contains the point that we want to detect. In Facial Expression Recognition Systems, only particular regions of the face are used for discrimination. The areas of the eyes, eyebrows, mouth, and nose are the main features in any Facial Expression Recognition System. Some facial recognition algorithms perceive facial components by extracting landmarks, or elements, from a picture of the subject's face. For instance, an algorithm may assess the relative position, size, and state of the eyes, nose, cheekbones, and jaw. These features are then used to search for other images with identical features. The majority of the facial expression recognition methods reported yet are focused on recognition of six primary expression categories such as: happiness, sadness, fear, anger, disgust and grief.

## b. Model Building and Training

The model used is convolution neural network with 3 layers, activation function to be ReLU.

Model: "sequential_1"		
Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 64)	1664
conv2d_2 (Conv2D)	(None, 48, 48, 64)	102464
batch_normalization_1 (Batch Normalization)	(None, 48, 48, 64)	256
activation_1 (Activation)	(None, 48, 48, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 64)	0
conv2d_3 (Conv2D)	(None, 24, 24, 128)	204928
conv2d_4 (Conv2D)	(None, 24, 24, 128)	409728
batch_normalization_2 (Batch Normalization)	(None, 24, 24, 128)	512
activation_2 (Activation)	(None, 24, 24, 128)	0
max_pooling2d_2 (MaxPooling2D)	(None, 12, 12, 128)	0
conv2d_5 (Conv2D)	(None, 12, 12, 256)	295168
conv2d_6 (Conv2D)	(None, 12, 12, 256)	590080
batch_normalization_3 (Batch Normalization)	(None, 12, 12, 256)	1024
max_pooling2d_3 (MaxPooling2D)	(None, 6, 6, 256)	0
flatten_1 (Flatten)	(None, 9216)	0
dense_1 (Dense)	(None, 6)	55302
Total params: 1,661,126		
Trainable params: 1,660,230		
Non-trainable params: 896		

**Fig1: Model Summary**

The model is trained for 22 epochs.

```
WARNING:tensorflow:From D:\ProgramData\Anaconda3\envs\tf\lib\site-packages\keras\backend\tensorflow_backend.py:422: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.

Train on 19204 samples, validate on 4802 samples
Epoch 1/22
19204/19204 [=====] - 941s 49ms/step - loss: 2.1196 - accuracy: 0.2506 - val_loss: 3.1224 - val_accuracy: 0.1120
Epoch 2/22
19204/19204 [=====] - 1029s 54ms/step - loss: 1.6831 - accuracy: 0.3419 - val_loss: 1.9673 - val_accuracy: 0.2820
Epoch 3/22
19204/19204 [=====] - 1022s 53ms/step - loss: 1.5141 - accuracy: 0.4075 - val_loss: 1.2800 - val_accuracy: 0.5104
Epoch 4/22
19204/19204 [=====] - 1021s 53ms/step - loss: 1.3980 - accuracy: 0.4606 - val_loss: 1.3254 - val_accuracy: 0.4708
Epoch 5/22
19204/19204 [=====] - 1327s 69ms/step - loss: 1.3135 - accuracy: 0.4903 - val_loss: 1.6442 - val_accuracy: 0.3419
Epoch 6/22
19204/19204 [=====] - 2481s 129ms/step - loss: 1.2217 - accuracy: 0.5231 - val_loss: 1.2263 - val_accuracy: 0.5217
Epoch 7/22
19204/19204 [=====] - 2468s 128ms/step - loss: 1.1615 - accuracy: 0.5526 - val_loss: 1.1267 - val_accuracy: 0.5562
Epoch 8/22
19204/19204 [=====] - 2479s 129ms/step - loss: 1.0884 - accuracy: 0.5824 - val_loss: 1.2823 - val_accuracy: 0.4990
Epoch 9/22
19204/19204 [=====] - 963s 50ms/step - loss: 1.0121 - accuracy: 0.6200 - val_loss: 1.5503 - val_accuracy: 0.4856
```

*Fig 2: The epochs*

### c. Methodology

#### Image Acquisition

Images used for facial expression recognition are static images. To take the images of expressions of people we use a Panasonic camera (Model DMC- LS5) with focal length of 5mm is used. The format of images is 24 bit color JPEG with resolution of 4320x 3240 pixels. The distance between the camera and person was four feet and images of six basic expressions of each person were taken.

#### Image Preprocessing

The image preprocessing procedure comes as a very important step in the facial expression recognition task. The objective of the preprocessing phase is to take images which have normalized intensity, uniform size and shape, and represent only a face expressing certain emotion. The preprocessing procedure should also reduce the effects of illumination and lighting. Expression representation can be delicate to translation, scaling, and rotation of the head in a picture. To battle the effect of these pointless changes, the facial image may be geometrically institutionalized before classification.

#### Feature Extraction

In developing accurate facial expression recognition system feature extraction is the most important stage. Unprocessed facial images hold vast amounts of data and feature extraction is

required to decrease it to smaller sets of data called features. Feature extraction change pixel information into a more elevated amount representation of color shape, motion, texture, and spatial configuration of the face or its features. The separated representation is utilized for further expression categorization. Feature extraction ordinarily decreases the information's dimensionality space. The reduction procedure ought to keep up essential data having high segregation force and high security.

### **Feature Selection**

Feature selection is concerned with choosing of a subset of features perfectly necessary to perform the classification task from a larger set of candidate features. The feature selection step has an effect on both the computational complexity and the quality of the classification results. It is essential that the information contained in the selected features is adequate to correctly verify the input class. Too many features may unnecessarily raise the complexity of the training and classification tasks, while a poor, inadequate selection of features may have a detrimental effect on the classification results. The process of selecting a sub set of features improves the efficiency of classifier and reduces execution time.

### **Classification**

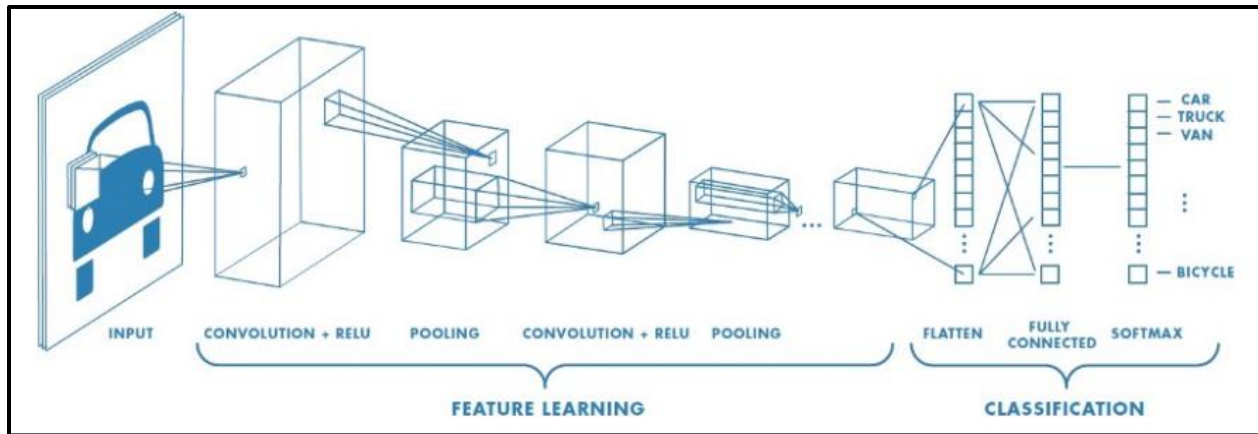
The last step of Facial Expressions Recognition systems is to recognize facial expression based on the extracted features. Classification refers to an algorithmic approach for recognizing a given expression as one of a given number of expressions. We use K- Nearest Neighbor classifier for classification. The KNearest Neighbor algorithm is a non-parametric method used for classification and regression. The input comprises of K closest training examples in the feature space. The output is class participation. By a majority vote of its neighbors an object is classified, with the object being allotted to the class most common among its k nearest neighbors.

### **Physically Data Set**



*Fig3: The Dataset*

#### **d. Convolution Neural Network**



*Fig4: CNN Model*

In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

CNN image classifications takes an input image, process it and classify it under certain categories (Eg., Dog, Cat, Tiger, Lion). Computers sees an input image as array of pixels and it depends on the image resolution. Based on the image resolution, it will see  $h \times w \times d$  ( $h$  = Height,  $w$  = Width,  $d$  = Dimension ). Eg., An image of  $6 \times 6 \times 3$  array of matrix of RGB (3 refers to RGB values) and an image of  $4 \times 4 \times 1$  array of matrix of grayscale image.

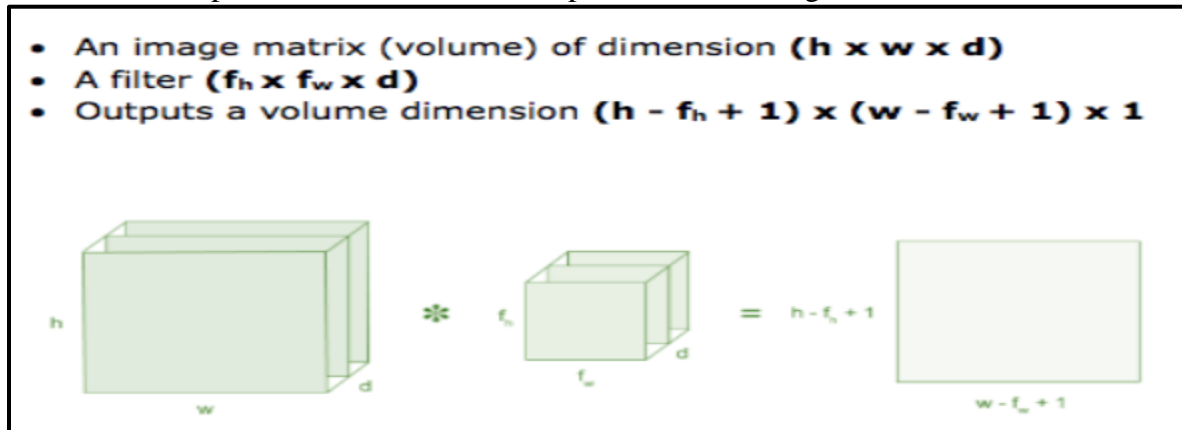
Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.

### **Convolution Layer**

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a

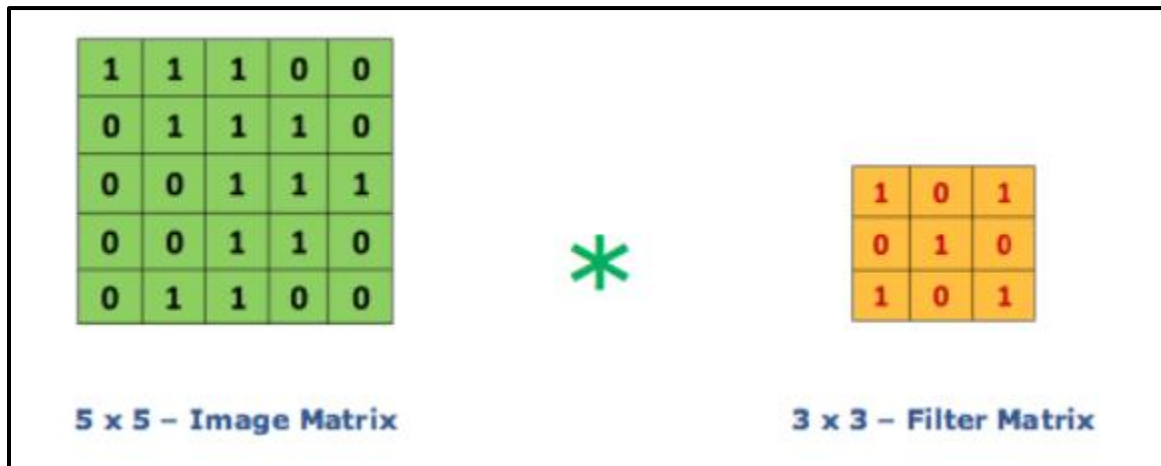


mathematical operation that takes two inputs such as image matrix and a filter or kernel.



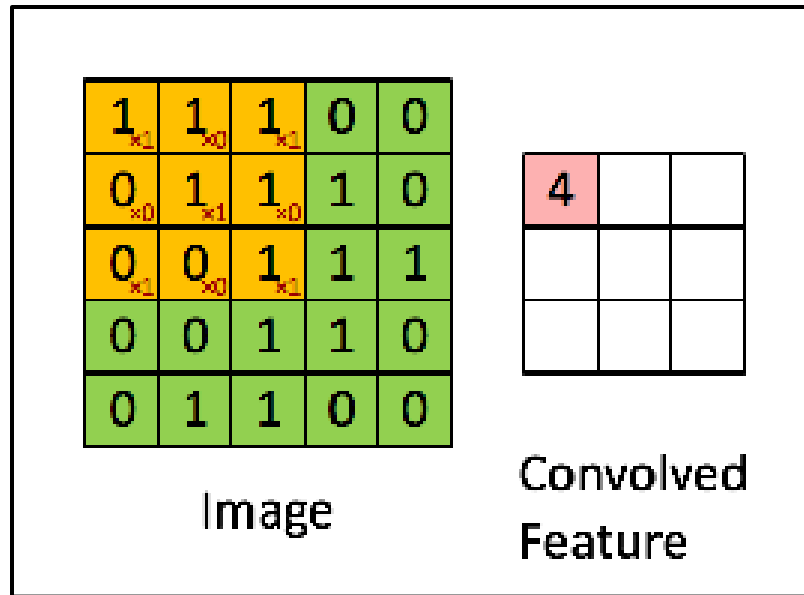
**Fig5: Convolution Layer**

Consider a 5 x 5 whose image pixel values are 0, 1 and filter matrix 3 x 3 as shown in below



**Fig6: Filter Matrix**

Then the convolution of 5 x 5 image matrix multiplies with 3 x 3 filter matrix which is called “Feature Map” as output shown in below

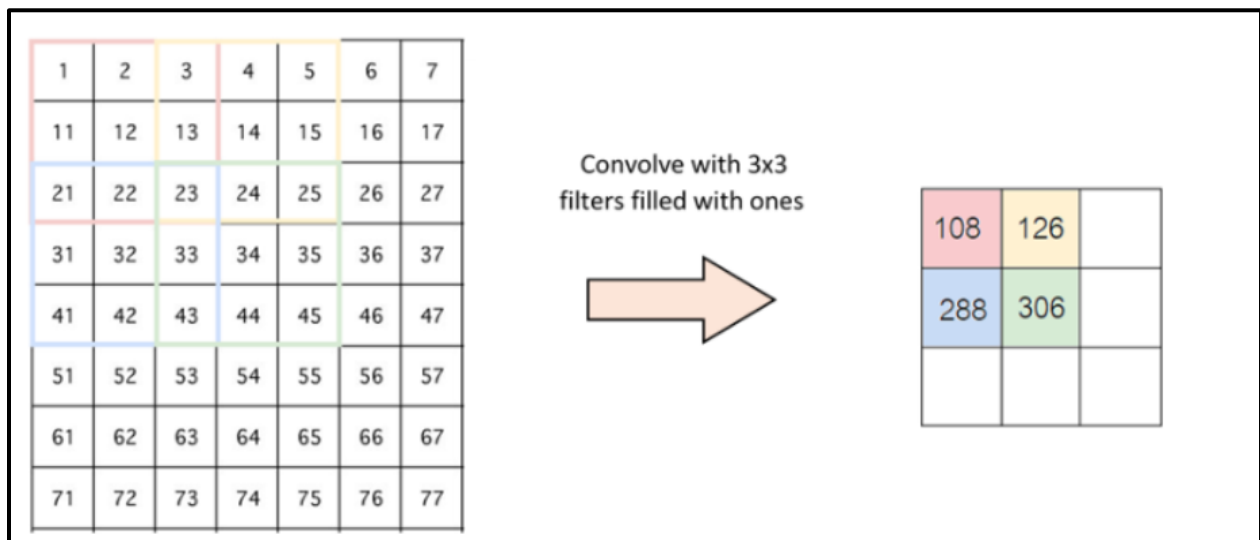


*Fig7: Convolved Feature*

Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters. The below example shows various convolution image after applying different types of filters (Kernels).

### Strides

Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on. The below figure shows convolution would work with a stride of 2.



*Fig8: Strides*

## Padding

Sometimes filter does not fit perfectly fit the input image. We have two options:

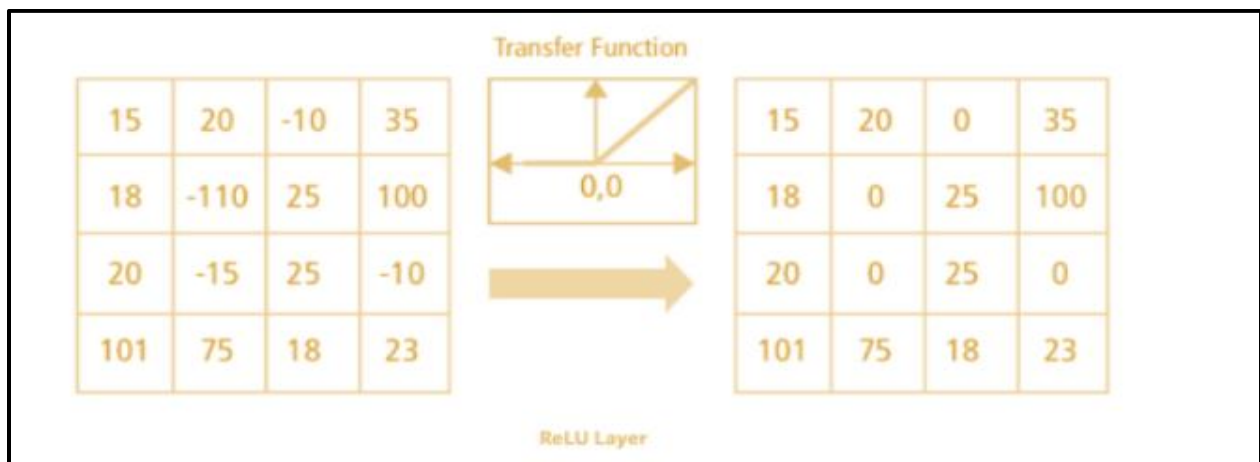
Pad the picture with zeros (zero-padding) so that it fits

Drop the part of the image where the filter did not fit. This is called valid padding which keeps only valid part of the image.

## Non Linearity (ReLU)

ReLU stands for Rectified Linear Unit for a non-linear operation. The output is  $f(x) = \max(0, x)$ .

Why ReLU is important : ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values.



**Fig9: ReLU**

There are other non linear functions such as tanh or sigmoid that can also be used instead of ReLU. Most of the data scientists use ReLU since performance wise ReLU is better than the other two.

## Pooling Layer

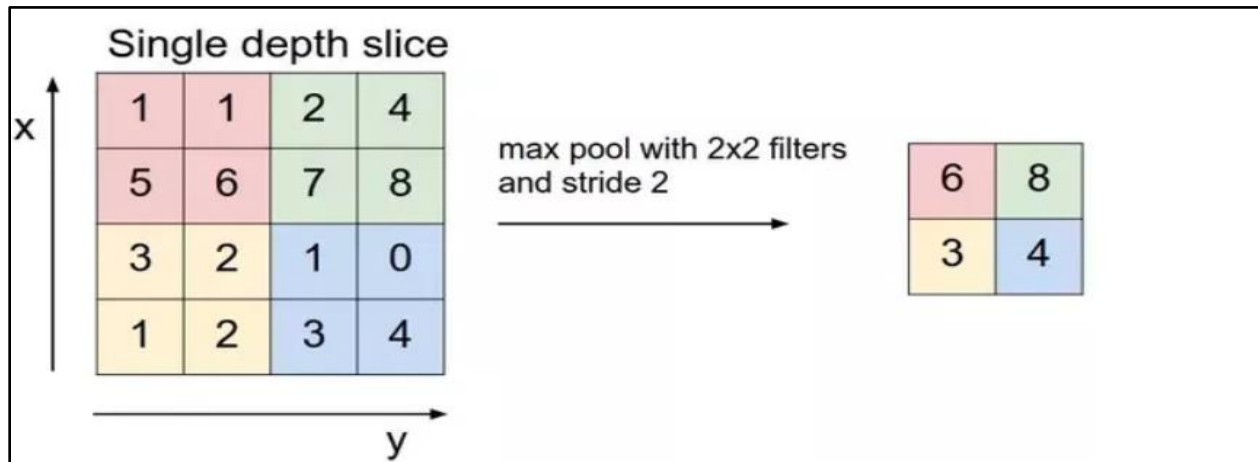
Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or downsampling which reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types:

Max Pooling

Average Pooling

Sum Pooling

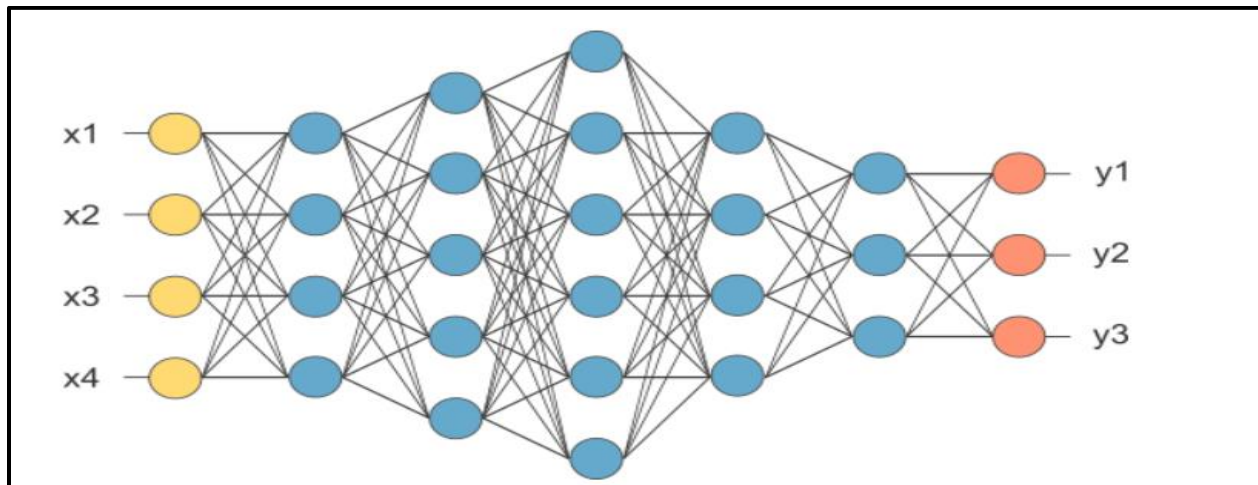
Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.



**Fig10: Pooling**

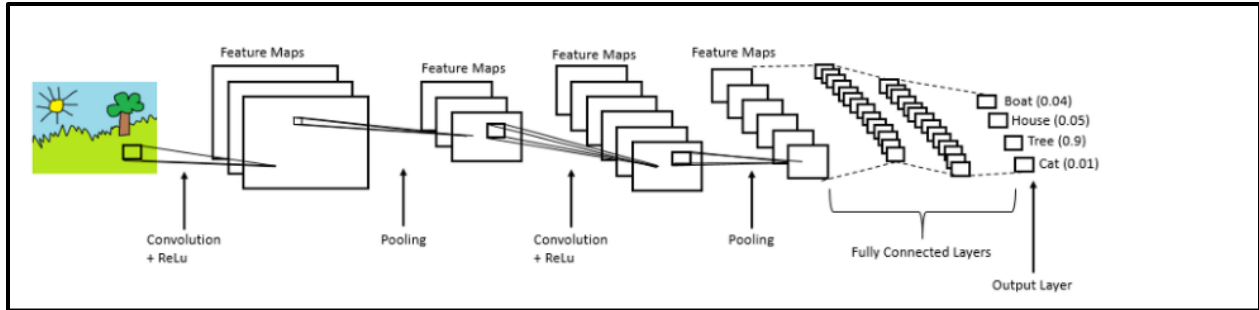
### Fully Connected Layer

The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like a neural network.



**Fig11: Last Layer**

In the above diagram, the feature map matrix will be converted as vector (x1, x2, x3, ...). With the fully connected layers, we combined these features together to create a model. Finally, we have an activation function such as softmax or sigmoid to classify the outputs as cat, dog, car, truck etc.,



*Fig12: The whole CNN*

#### e. Data

For this project, a Facial Expression Recognition dataset, fer2013.csv, is used to train the model and predict the facial expression into 7 broad categories as mentioned in the objective.

The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is centered and occupies about the same amount of space in each image.

There are majorly 2 columns in the dataset:

1. Emotion: The "emotion" column contains a numeric code ranging from 0 to 6, inclusive, for the emotion that is present in the image.
2. Pixel: The "pixels" column contains a string surrounded in quotes for each image. The contents of this string a space-separated pixel values in row major order.

### III. Solution Development

#### a. Back-end and Front-end

##### Back-end

The coding was done in python using a convolution neural network of 3 layers with the following specifications:

	#Filters	Activations Function	Normalisation	Filter dimension	padding	pool size
First Layer	64	Relu	Batch Normalization	5*5	same	2*2

<b>Second Layer</b>	128	Relu	Batch Normalization	5*5	same	2*2
<b>Third Layer</b>	256	Relu	Batch Normalization	3*3	same	2*2

***Table1: CNN***

The last layer uses the softmax regression to flatten the image and the “categorical\_crossentropy” is used as a loss function with “adam” as an optimizer. The model is trained for 22 epochs and saved for it to predict the expression of the image uploaded from the front-end.

### **Front-end**

The front end was developed using HTML with 3 files:

1. index.html: This is the home page that appears and expects the user to upload an image and submit it.
2. App.py: this file receives the image and then converts it to grayscale, crops it and creates a bounding box around the face to accurately predict the expression. Then the trained mode is loaded to predict the transformed image and the results are sent another html file.
3. After.html: This html contains the result of the expression deduced by the model along with the finally transformed image.

The app is finally deployed on Docker.

### **b. Software and Hardware Requirements**

<b>Frontend UI</b>	<b>Backend programming</b>
HTML	Python
CSS	
Docker	
<b>Software</b>	<b>Framework</b>
Spyder	Flask
Jupyter Notebook	

***Table2: Requirements***

### **c. Deploying it on docker**

1. Creating the DockerFile that runs the app.py

```

#Create a ubuntu base image with python 3 installed.
FROM python:3

#Set the working directory
WORKDIR /usr/src/app

#copy all the files
COPY . .

#Install the dependencies
RUN apt-get -y update
RUN pip install --upgrade pip
RUN pip install -r requirements.txt
RUN apt-get update ##[edited]
RUN apt-get install ffmpeg libsm6 libxext6 -y

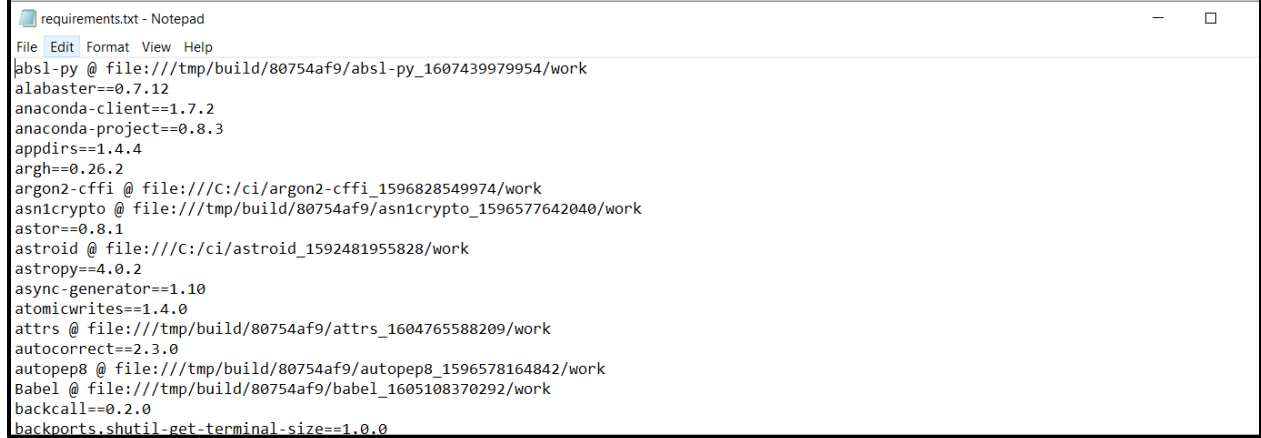
#Expose the required port
EXPOSE 5003

#Run the command
CMD ["python3", "./app.py"]

```

*Fig13: The app.py*

2. Creating a requirements.txt file that contains the required packages.



```

requirements.txt - Notepad
File Edit Format View Help
absl-py @ file:///tmp/build/80754af9/absl-py_1607439979954/work
alabaster==0.7.12
anaconda-client==1.7.2
anaconda-project==0.8.3
appdirs==1.4.4
argh==0.26.2
argon2-cffi @ file:///C:/ci/argon2-cffi_1596828549974/work
asn1crypto @ file:///tmp/build/80754af9/asn1crypto_1596577642040/work
astor==0.8.1
astroid @ file:///C:/ci/astroid_1592481955828/work
astropy==4.0.2
async-generator==1.10
atomicwrites==1.4.0
attrs @ file:///tmp/build/80754af9/attrs_1604765588209/work
autocorrect==2.3.0
autopep8 @ file:///tmp/build/80754af9/autopep8_1596578164842/work
Babel @ file:///tmp/build/80754af9/babel_1605108370292/work
backcall==0.2.0
backports.shutil-get-terminal-size==1.0.0

```

*Fig14: Requirements.txt*

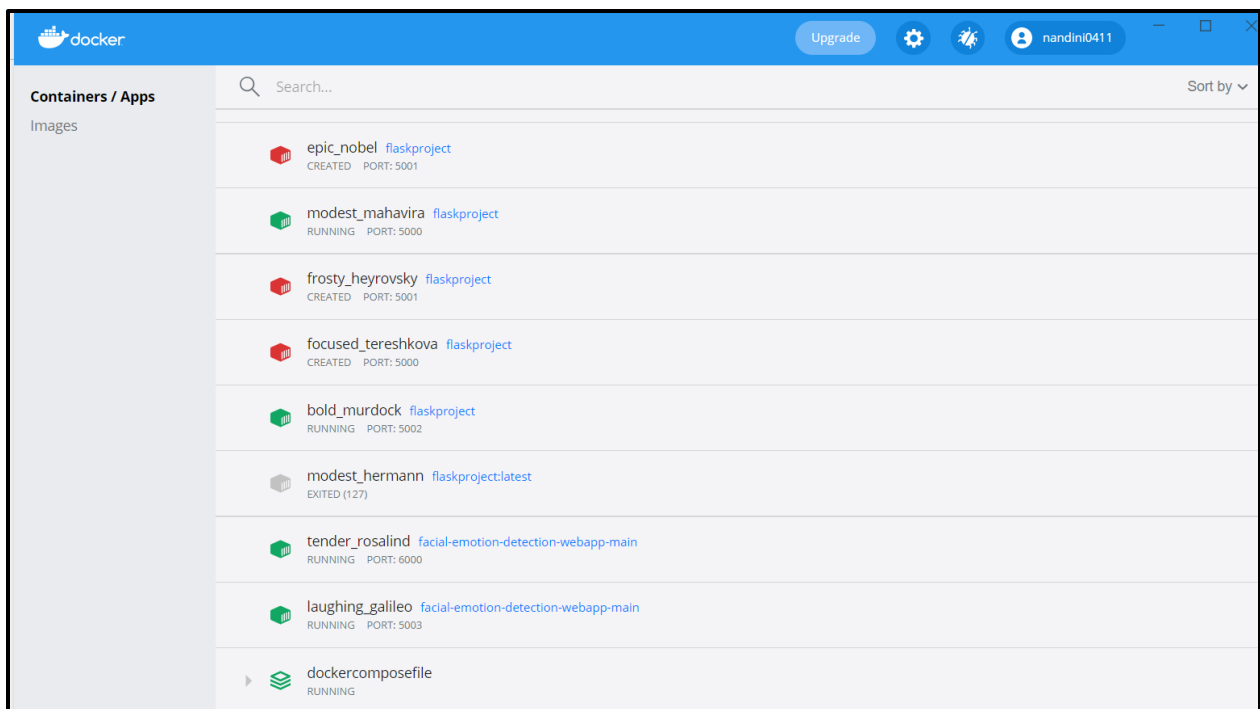
3. Running the following command on command prompt

**docker build -t flaskproject .**

```
(tf) C:\Users\nandi\AI Enterprise\FP\facial-emotion-detection-webapp-main\facial-emotion-detection-webapp-main>docker build -t facial-emotion-detection-webapp-main .
[+] Building 23.0s (10/10) FINISHED
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 380B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/python:3 0.7s
=> [internal] load build context 0.1s
=> => transferring context: 98.77kB 0.0s
=> [1/5] FROM docker.io/library/python:3@sha256:438cb846732e397ec5d94b1aea461fe26a51b901d510ca105eba5595621db3fe 0.0s
=> CACHED [2/5] WORKDIR /usr/src/app 0.0s
=> [3/5] COPY . . 2.5s
=> [4/5] RUN apt-get -y update 7.0s
=> [5/5] RUN pip3 install -r requirements.txt 9.5s
=> exporting to image 2.7s
=> => exporting layers 2.6s
=> => writing image sha256:f25fed53cbeca6a5aa233f10fb110637a208c6daba4b50fc92e485b85d32821d 0.0s
=> => naming to docker.io/library/facial-emotion-detection-webapp-main 0.0s
```

**Fig15: Building container**

#### 4. Verifying that a container is created on the docker



**Fig16: Docker dashboard**

#### 5. Running the application on the docker

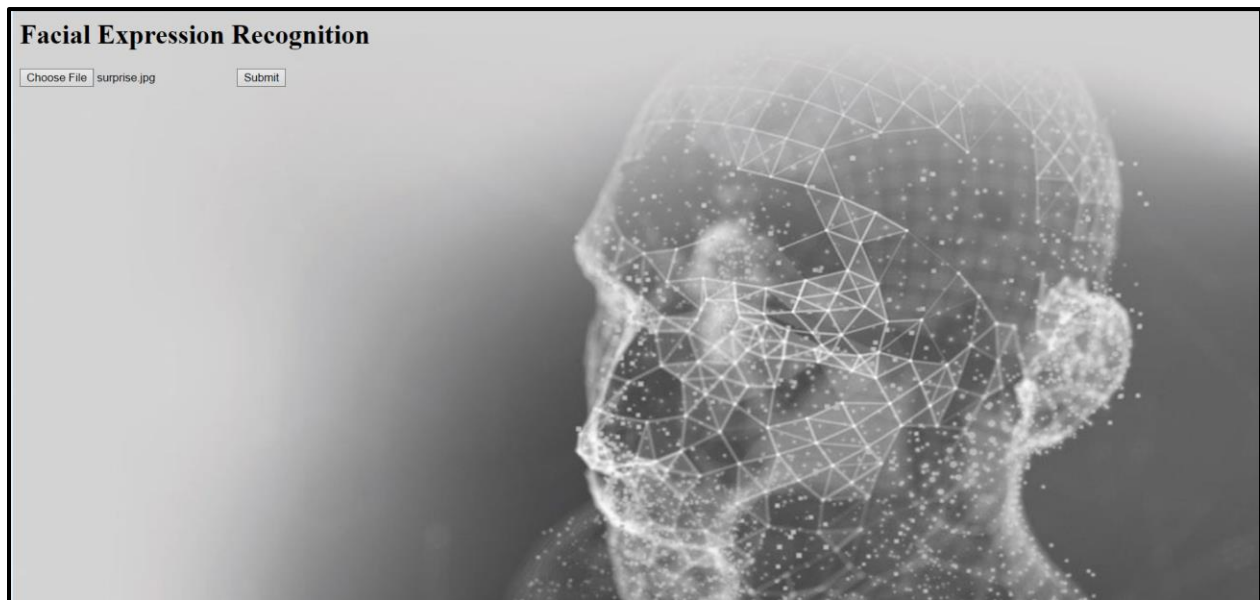


```
docker exec -it fb8b835ee9a14a8dc3c4c343533f283897c1cc955084ca4142d82f4c657e6685 /bin/sh
# python app.py
2021-04-07 04:47:42.395705: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic library 'libcudart.so.11.0'; dlderror: libcudart.so.11.0: cannot open shared object file: No such file or directory
2021-04-07 04:47:42.395774: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
* 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
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
2021-04-07 04:47:46.199700: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic library 'libcudart.so.11.0'; dlderror: libcudart.so.11.0: cannot open shared object file: No such file or directory
2021-04-07 04:47:46.199785: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
* Debugger is active!
* Debugger PIN: 326-768-354
```

***Fig17: Running the application on docker***

## 6. Launching the app in the browser

### Uploading an image

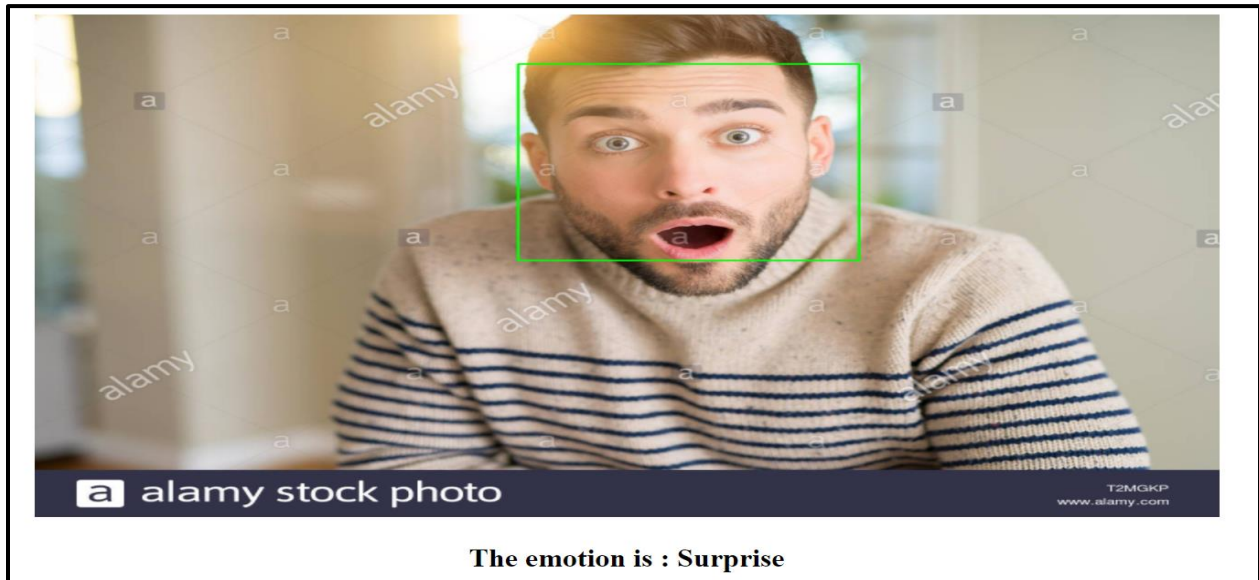


***Fig18: Index.html***

### Predicting the expression

A bounding box is created around the face of the object which makes it easier to predict the expression.

- A surprised expression



*Fig19: Surprised expression*

- A scared expression



*Fig20: Scared expression*

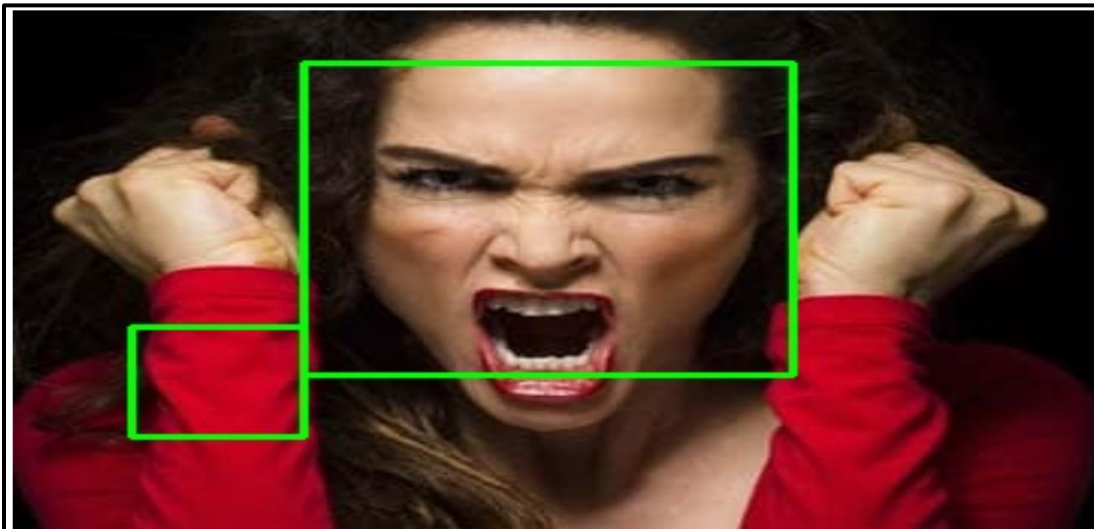
- A happy expression



**The emotion is : Happy**

*Fig21: Happy expression*

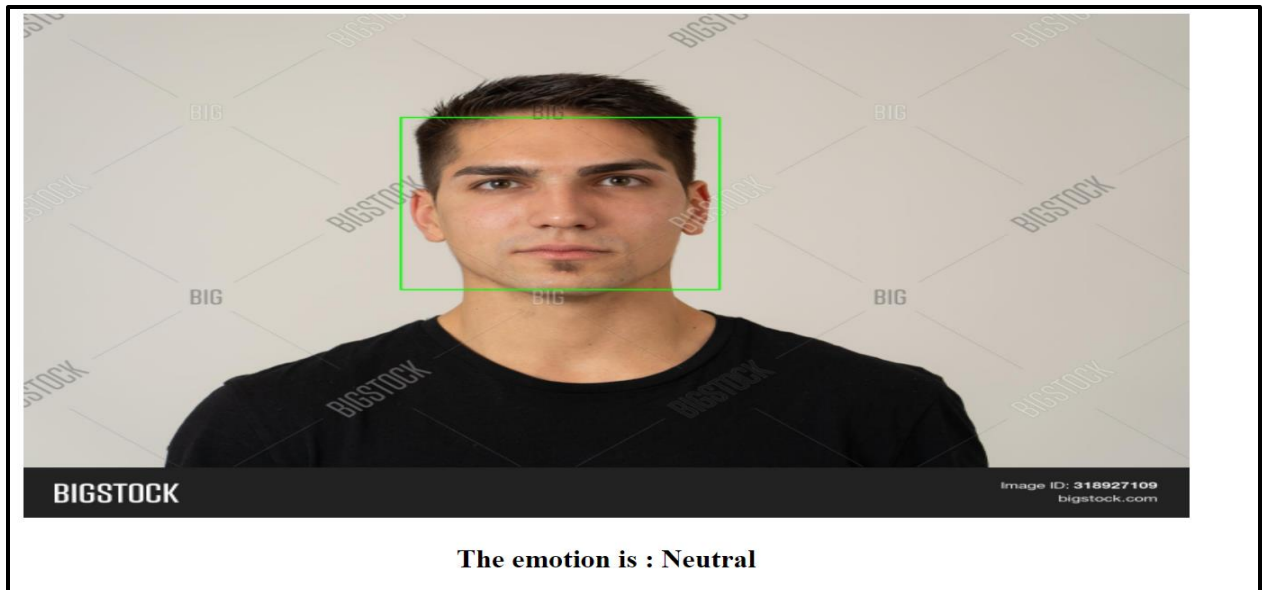
- An angry expression



**The emotion is : Anger**

*Fig22: Angry expression*

- A neutral Expression



*Fig23: Neutral expression*

#### IV. Project Timelines

Project Key Activities	Week					
	1	2	3	4	5	6
	March			April		
	8	15	22	29	5	12
1.0 Project Proposal and Presentation						
1.1 Submission of Initial Project Proposal						
1.2 Presentation of Project proposal presentation						
2.0 Preliminary Analysis and EDA						
2.1 Creation of Preliminary Analysis						
2.2 Preliminary Analysis Exploratory Data Analysis (EDA)						
3.0 Preliminary Report						
3.1 Produce Preliminary Report						
4.0 Final Project Report and Presentation						
4.1 Submission of Final Project Report						
4.2 Submission of Final project Presentation video						
4.3 Presentation of Final Project						

*Table3: Deliverables*

#### V References

1. Nazia Perveen, Nazir Ahmad, M. Abdul Qadoos Bilal Khan, Rizwan Khalid, Salman Qadri, 2016, "Facial Expression Recognition Through Machine Learning"
2. <https://www.tutorialspoint.com/build-and-deploy-a-flask-application-inside-docker>

3. [https://www.youtube.com/watch?v=QBOcKdh-fwQ&list=RDCMUCTt7pyY-o0eltq14glaG5dg&start\\_radio=1&t=9&ab\\_channel=AutomationStepbyStep-RaghavPal](https://www.youtube.com/watch?v=QBOcKdh-fwQ&list=RDCMUCTt7pyY-o0eltq14glaG5dg&start_radio=1&t=9&ab_channel=AutomationStepbyStep-RaghavPal)
4. Prabhu, 2018, “Understanding of Convolutional Neural Network (CNN) — Deep Learning”