Journal of Physics: Conference Series

PAPER • OPEN ACCESS

A real time face emotion classification and recognition using deep learning model

To cite this article: Shaik Asif Hussain and Ahlam Salim Abdallah Al Balushi 2020 *J. Phys.: Conf. Ser.* **1432** 012087

A real time face emotion classification and recognition using deep learning model

Dr. Shaik Asif Hussain, Ahlam Salim Abdallah Al Balushi

Department of Electronics and Communication Engineering, Middle East College, Muscat.

Email**:** sah.ssk@gmail.com

Abstract. Facial Detection and recognition research has been widely studied in recent years. The facial recognition applications plays an important role in many areas such as security, camera surveillance, identity verification in modern electronic devices, criminal investigations, database management systems and smart card applications etc. This work presents deep learning algorithms used in facial recognition for accurate identification and detection. The main objective of facial recognition is to authenticate and identify the facial features. However, the facial features are captured in real time and processed using haar cascade detection. The sequential process of the work is defined in three different phases where in the first phase human face is detected from the camera and in the second phase, the captured input is analyzed based on the features and database used with support of keras convolutional neural network model. In the last phase human face is authenticated to classify the emotions of human as happy, neutral, angry, sad, disgust and surprise. The proposed work presented is simplified in three objectives as face detection, recognition and emotion classification. In support of this work Open CV library, dataset and python programming is used for computer vision techniques involved. In order to prove real time efficacy, an experiment was conducted for multiple students to identify their inner emotions and find physiological changes for each face. The results of the experiments demonstrates the perfections in face analysis system. Finally, the performance of automatic face detection and recognition is measured with Accuracy.

1. Introduction

Human computer interaction is a common trend and innate ability to distinguish among multiple faces. Until recent past computer vision problems were quite challenging but advent of modern technologies has trivially improved from the problems of varying light, changed by age, hair and other accessories [1]. However, face recognition applications are used improve access to identify and verify the people by their face features. Hence interpreting the facial features and their actions is much required. As these features and expressions helps in classify the emotions of human face. Recent advances in technology has resulted in the use of Artificial intelligence system as these systems are capable to understand and realize the emotion recognition through facial features. Hence this is an attempt to prove the existence of latest technological developments for human-computer interaction using deep learning or Convolution neural network models [2]. To recognize and classify the human face various methods are required but deep learning technique outperforms other methods by its large capabilities of different datasets and fast computation capabilities. Usually the process of face recognition and classification involves various steps such as preprocessing, detection, orientation, extraction of features and classification of emotion. These tasks are easily performed with deep learning keras model which outperforms the handy computations. Deep learning technique is a standard paradigm to represent the working of human brain with neurons [3]. This learning usually consists of neural network model where neurons act as inputs and each of them are connected to move as outputs. Deep learning is the subset of machine learning in which the algorithms are integrated similar to machine learning, but there are innumerable levels of these algorithms, each of which provides a different interpretation of the data they insert. This network of algorithms is called the network of artificial neurons, because their operation is a source of inspiration, or we could say; an attempt to mimic the function of human neural networks in the brain [7]. Several hidden levels allow deep neural networks to study data functions in the so-called functional hierarchy, since simple functions, such as two pixels, combine from one level to another, forming, for example, functions that are more complex. Low level networks are not much capable of processing mathematical operations than multilevel networks as these networks are able to perform deep data processing. Deep learning models have very good characteristics of calculating the intensive calculus and has great demand in the form of minute chip called Graphical processing unit [9].There are three different phases of this work such as facial detection, recognition and emotion classification which are outlined briefly in the below sections.

1.1 Convolution Neural Networks

The development of Convolution neural network has a significant impact on the computer vision domain in recent years in addition to this it has an important step and ability to recognize objects [11]. This gaining importance of the technique was possible because of increasing value of the computer and the amount of data available to form a neural network.

Convolutional neural network (CNN) models have kernels to detect border functions or outline for an image. This model has weights arranged in array of values to form and obtain desired characteristics. Every CNN model allocates space to determine the control of image to be recognized. The values in the image represents a degree to which convolution operation depends, hence the product is calculated and determined with the location in the image [13].

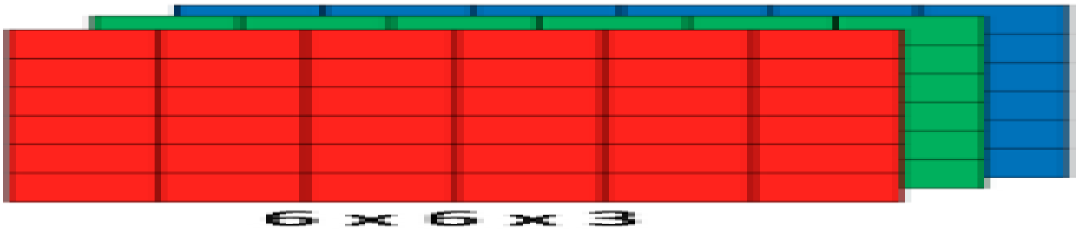


Figure 1. Array of RGB Matrix

The CNN model is aimed to train and test the designed system. Here the image is an input with several layers as filters connected together with weights as desired. It has Softmax function used to obtain and classify the image between 0 and 1. Hence the below figure 2 shows the representation of CNN model processing based on the input image.

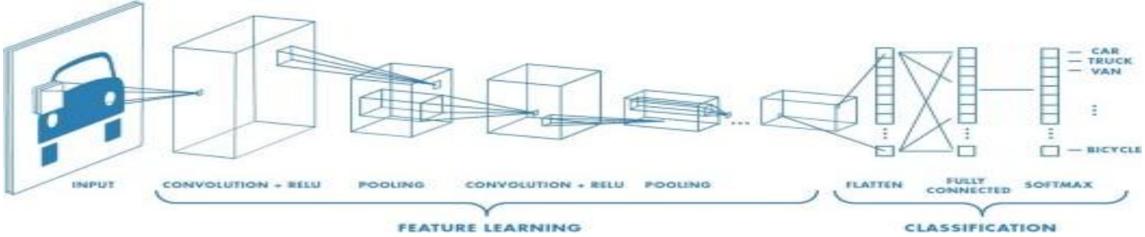


Figure 2. Neural network with many convolutional layers

Consider below figure 3 shows a basic first layer of CNN model where features are extracted from input image. After extracting the features they are passed to convolution layer where weights functions are multiplied to obtain the product of each layer. The mathematical processing takes place between image matrix, 2 inputs and kernel values. The obtained image matrix represented with dimensions and volume is given as (h x w x d) in figure 1and the filter as (f_h x f_w x d) and outputs of volume dimension is given as (h-f_h+1) x (w-f_w+1) x 1. Further details of network layers, weights and function values are described in section II.

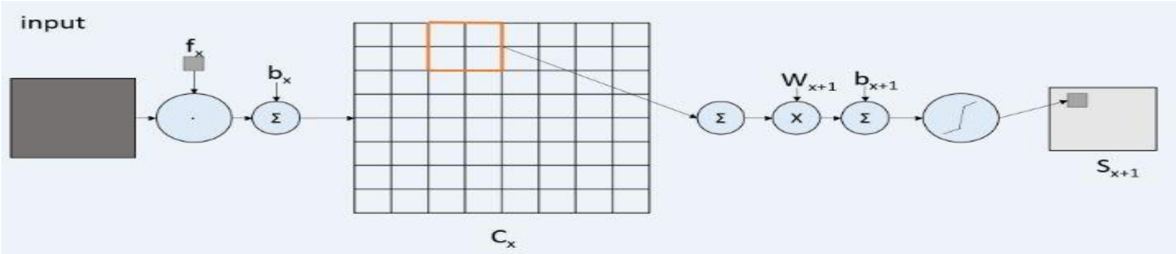


Figure 3. Main Process of CNN

The most important technology in CNN is the local field of perception, weight distribution, subsampling of time or space to separate functions and reduce the size of training parameters. The advantage of the CNN algorithm is that it avoids a clear separation of functions and indirect learning of training data. The use of the subsample structure in time or space can provide stability, size and speed; Network entry and topology can be very good and offer the unique benefits of voice recognition and image processing. (Tianyi Liu, et al., 2015). CNN algorithm requires experience in project architecture and must constantly recognize application practice errors in order to get the best solution for a specific CNN application architecture. Based on an input gray value of 96*96 at the pre-processing stage, which provides 32. (Tianyi Liu, et al., 2015)

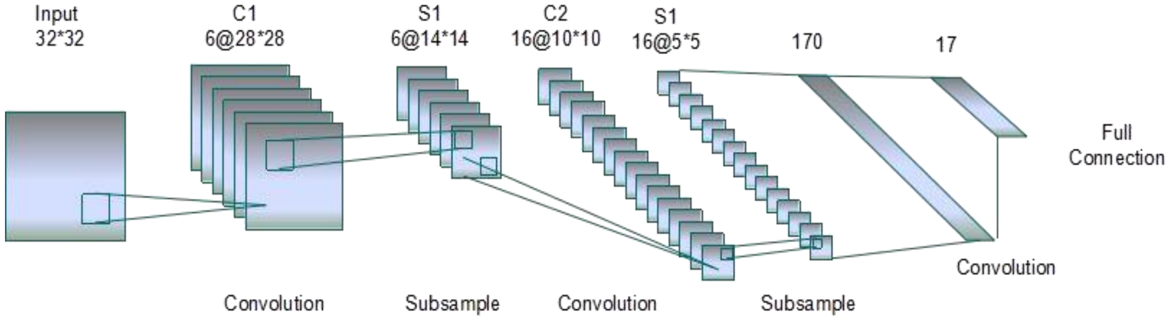


Figure 4. Convolution neural network (CNN) model stages

2. Approach

The proposed work carries out in three sequential steps as Face Detection, Face Recognition and Face Classification. In the first step a video camera is used to capture the human face and detect the exact location of face by a bounding box coordinates for the face detected in real-time. This step involves face detection using Haar cascade detection with open CV library. Viola jones algorithm and haar cascade features are combined to detect human face. The images detected have shapes, objects and landscapes etc. In this phase human face is detected and face features are extracted and stored in the database for face recognition. The CNN model as shown in figure 4 uses VGG 16 to match the face from the database and recognize with the name for the face detected. Faces are recognized from the database and are compared to identify or detect the face through embedding vectors. The distribution platform use Anaconda and python 3.5 software in processing face detection, recognition and classification. The image features in the database dlib and other libraries. First face is detected and then recognized with the database features and matching using CNN model training and testing database. Finally the recognized human face is classified based on the expression in real time as Angry, fear, disgust, happy, neutral and surprise. The network architecture VGG 16 is built with CNN model for large database recognition and classification. The designed network model has honey comb 3 x 3 layers where the two connected layers have 4096 nodes with Softmax classification. The local binary model histogram is used as open CV library for detecting the human faces. The image pixels are identified by setting a threshold and the end result is represented in form of a binary number. In order to perform this LBPH uses 4 parameters like radio, neighbors, Grid X and Y. Here the radio button used defines the local radius in circular manner for the binary model. The circular local binary model represents the number of sample points as neighbors. The X and Y dimensions of grid defines the number of horizontal and vertical direction cells. The circular binary model is formed to the marked areas with appropriate label to receive the test data from machine. To perform the procedural result of face detection first create a folder and name it and write two python files create_data.py face_recognize.py then haarcascade_frontalface_default.xml library set and run complete files from Integrated development environment of python [15].

2.1. Proposed Functional Design

The proposed system and its functioning is shown in the below figure 5. The individual functional blocks have their own specific operation. The image is captured using HP 3000 camera which is preprocessed further to feed the neural networks. The captured image is modelled using network architecture to build and train the dataset for identification and emotion classification. The details are enlisted in sequential steps as flow chart in figure 6

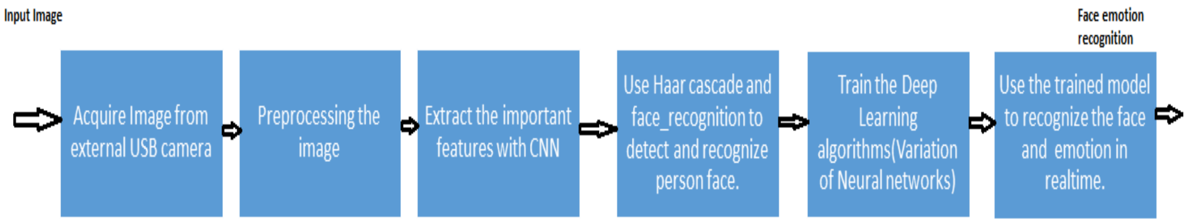

Input Image
Face emotion recognition
Acquire Image from external USB camera
Preprocessing the image
Extract the important features with CNN
Use Haar cascade and face_recognition to detect and recognize person face.
Train the Deep Learning algorithms(Variation of Neural networks)
Use the trained model to recognize the face and emotion in realtime.

Figure 5. Block diagram of Face detection and recognition classification

2.2. Flowchart and Design Specifications

The Real time face detection, recognition and classification is depicted in the figure 5. The captured image is bounded in a box and converted in to binary pattern to specify it as feature vector and stored in database. The images are trained to match the input image and also classify the expression of facial features as sad, angry, happy, disgust and neutral. In the training process, there are a total of seven steps which are represented below as loading dataset, preprocessing, augmenting the data as feature vector, building and compiling the design model, training and storing the feature vector and validating the test model.

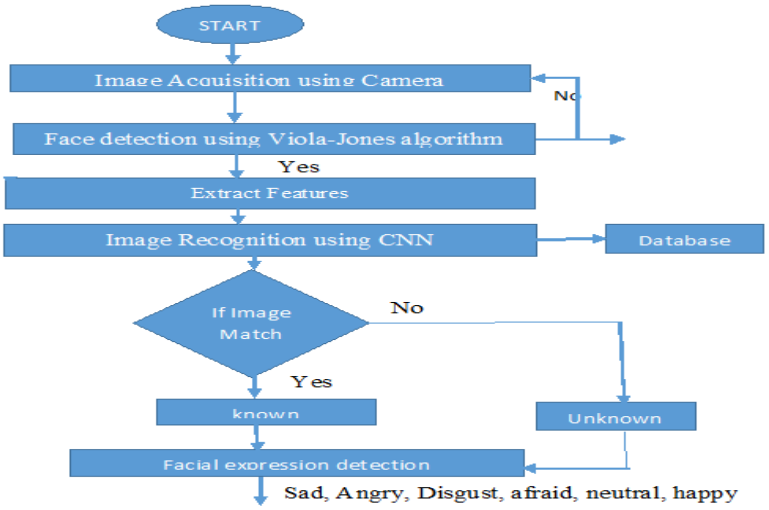

START
Image Acquisition using Camera
No
Face detection using Viola-Jones algorithm
Yes
Extract Features
Image Recognition using CNN
Database
If Image Match
No
Yes
known
Unknown
Facial expression detection
Sad, Angry, Disgust, afraid, neutral, happy

Figure 6. Face detection and recognition flow chart

The work involves local binary pattern histogram (LBPH) to convert the captured image in to binary vector. This processing helps in detection of face using viola jones algorithm. The image pixels are stored with defined threshold for facial feature vector. These vectors are added with weights to form a network architecture model for facial expression classification using VGG 16 CNN model as shown in figure 7. Usually VGG 16 is trained with image net database where 16 layers of network are classified in to 1000 categories. The 3x3 convolution layers are stacked on one another to measure max pooling. These network connections are related with 4096 nodes as softmax classifier.

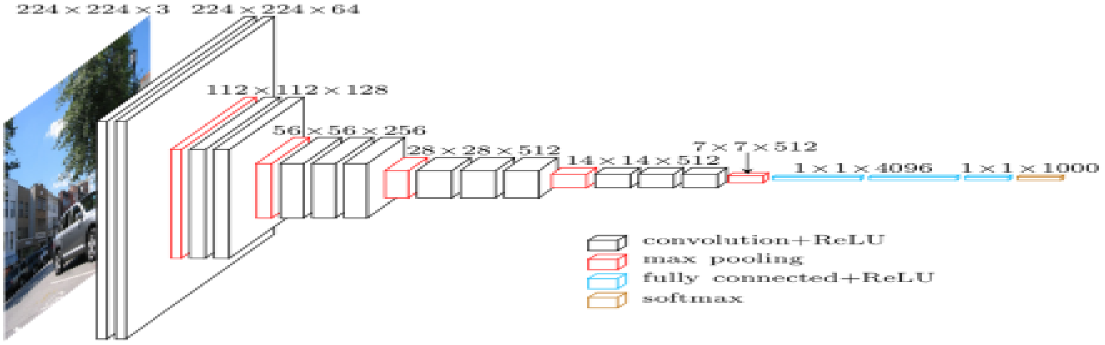


Figure 7. VGG 16 net model

3. Implementation

The system design model is tested functionally in four different levels

3.1. Building a dataset to classify emotion using convolution neural network layer

The free dataset used is proposed work is represented in the link (http://www.kdef.se/) given. The KDEF data set as in figure 8 had 4900 pictures where 1999 pictures were removed as they were in side postures and trained the dataset with 2901 pictures by splitting and training the dataset in to 70 % and 30% respectively. Similarly with my images own dataset was also created with 100 images.

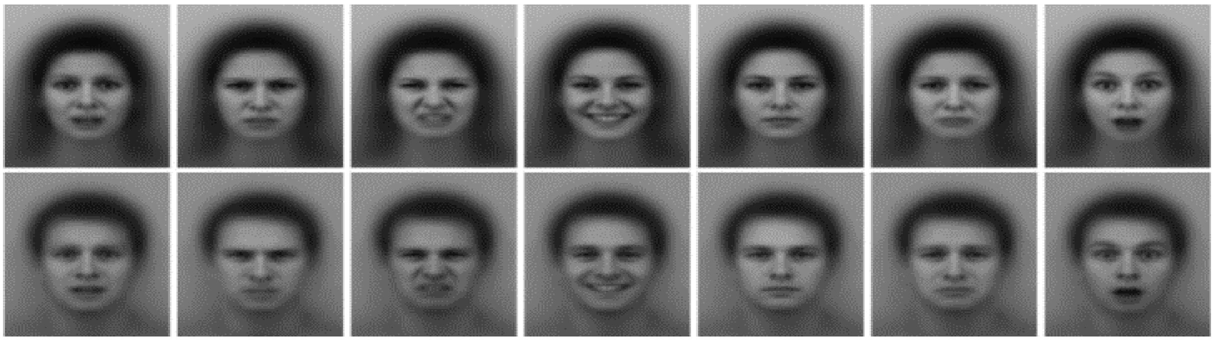
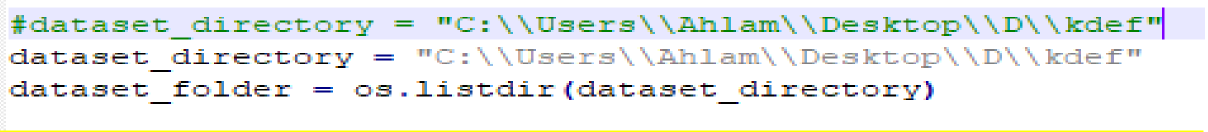


Figure 8. Sample KDEF data set (KDEF | The Kenya Dryland Education Fund 2019)

The trained dataset folder is



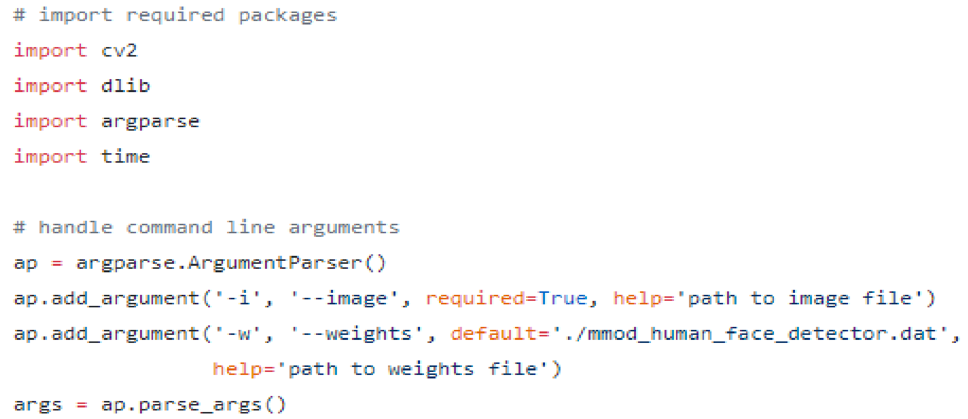
```
#dataset_directory = "C:\\Users\\Ahlam\\Desktop\\D\\kdef"
dataset_directory = "C:\\Users\\Ahlam\\Desktop\\D\\kdef"
dataset_folder = os.listdir(dataset_directory)
```

3.2. Creating database with open CV library and other resources

a) First make a folder and name it and then create two python files such as face_recognize.py and create_data.py. The code is copied in the resultant source file and executed to check the errors. Similarly copy the xml file in to project directory to support the face features. The file to be copied is haarcascade_frontalface_default.xml.

b) Dlib: It is cross platform library programmed using C++. It is used for face recognition and supports the face features using linear SVM and Histogram of oriented gradients (HoG).

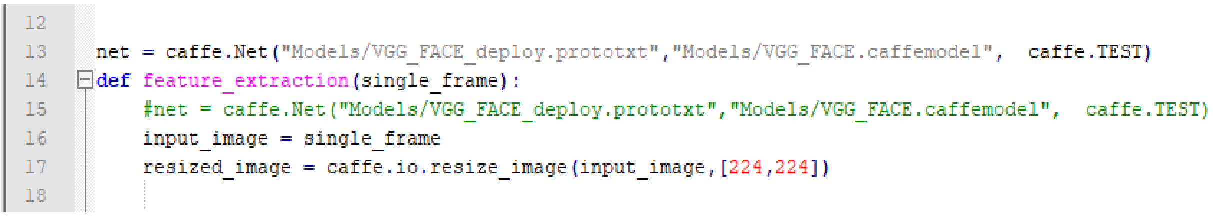
The Dlib imported to perform recognition phase



```
# import required packages
import cv2
import dlib
import argparse
import time

# handle command line arguments
ap = argparse.ArgumentParser()
ap.add_argument('-i', '--image', required=True, help='path to image file')
ap.add_argument('-w', '--weights', default='./mmod_human_face_detector.dat',
                help='path to weights file')
args = ap.parse_args()
```

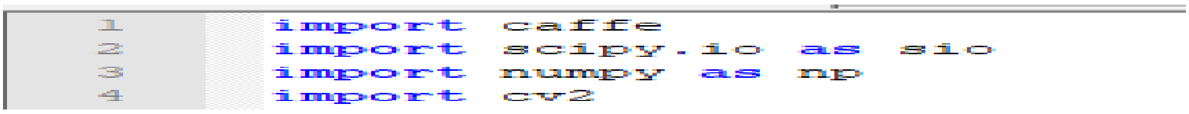
c) VGG caffe model: This model is imported to support and train Convolution neural network as solver and model definition, data preparation and training. This model is used to predict the unused data during python script writing. Caffe model in python programming for CNN is given below



```
net = caffe.Net("Models/VGG_FACE_deploy.prototxt","Models/VGG_FACE.caffemodel",  caffe.TEST)
def feature_extraction(single_frame):
    #net = caffe.Net("Models/VGG_FACE_deploy.prototxt","Models/VGG_FACE.caffemodel",  caffe.TEST)
    input_image = single_frame
    resized_image = caffe.io.resize_image(input_image,[224,224])
```

d) Get-pip, numpy and Pandas: This is open source library tool used as installation package in python programming language.

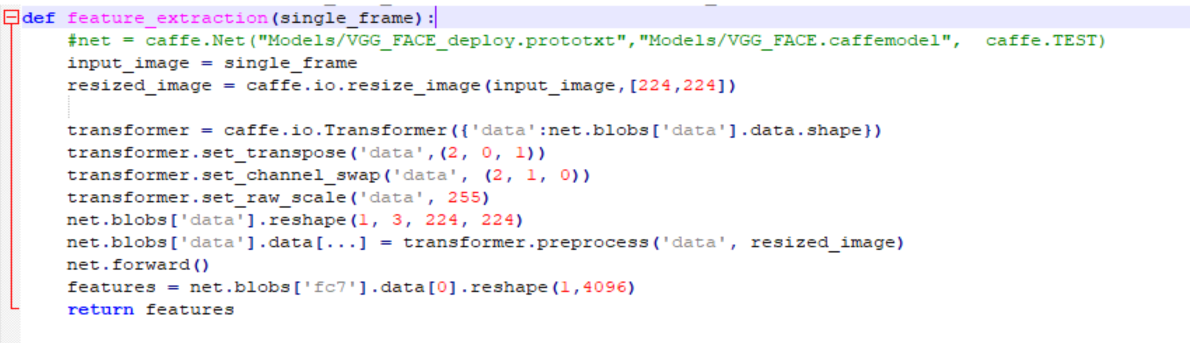
e) open CV, sciPy, SVM and pickle: These are open source library for computer vision, computation operations and feature vectors. Imported library functions for python 3.5



```
import caffe
import scipy.io as sio
import numpy as np
import cv2
```

3.3 Training procedure for recognition and classification

The training and testing data is splitted in to two models where VGG caffe model builds feature vector with support vector machine to train the emotion classification. The size of the image used is 224x 224 which is preprocessed as rescaling and extracted as feature vector to train the classifier. Main code for feature extracted to train the classifier is given below

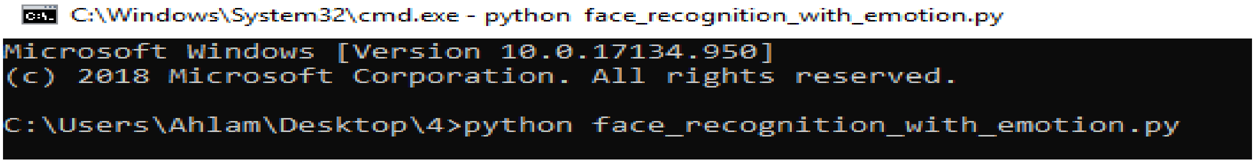


```
def feature_extraction(single_frame):
    #net = caffe.Net("Models/VGG_FACE_deploy.prototxt","Models/VGG_FACE.caffemodel",  caffe.TEST)
    input_image = single_frame
    resized_image = caffe.io.resize_image(input_image,[224,224])

    transformer = caffe.io.Transformer({'data':net.blobs['data'].data.shape})
    transformer.set_transpose('data',(2, 0, 1))
    transformer.set_channel_swap('data', (2, 1, 0))
    transformer.set_raw_scale('data', 255)
    net.blobs['data'].reshape(1, 3, 224, 224)
    net.blobs['data'].data[...] = transformer.preprocess('data', resized_image)
    net.forward()
    features = net.blobs['fc7'].data[0].reshape(1,4096)
    return features
```

3.4. Building all the files to Python 3.5 and Anaconda 3-4.2.0 software platform.

The images are embedded and generated by creating embeddings.py file and library of facial features are encoded using encoded_images_data.csv file. Face recognition with emotion python 3.5 is given below



```
C:\Windows\System32\cmd.exe - python face_recognition_with_emotion.py
Microsoft Windows [Version 10.0.17134.950]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Ahlam\Desktop\4>python face_recognition_with_emotion.py
```

The software platform used in processing the results is Python 3.5 and Anaconda 3-4.2.0.

4. Experimental Results

The results are followed in three phases as face detection, Face recognition and Face classification. Each phase is explained as below for better understanding. The ability to recognize real time human faces is done by using latest software and database built on computer technology. The Human face detection involves objects, landscapes and background etc. Facial recognition is built with CNN model where the facial features are extracted and classified as per VGG 16 model. The classification of human face is defined in various expressions like disgust, sad, happy, fear and neutral. Based on the classified expression matching is displayed on the LCD screen.

The work is carried out in three different phases where in the first phase the camera reads the image and captures it based on the openCV haar cascade detection and viola jones algorithm. The RGB image captured and processed with a database of extracted features for facial recognition. The image is converted in matrix with range of values 0 to 255. The training model of database features are tested to recognize the human face where the image resolution of 1024*1024*3 is reduced to 227*227*3 to perform matrix manipulations as per CNN model.

Face detection: In the first step testing of the face is detected with 1024 x 1024 resoulution using viola jones Algorithm as shown in figure 9 & 10 below.

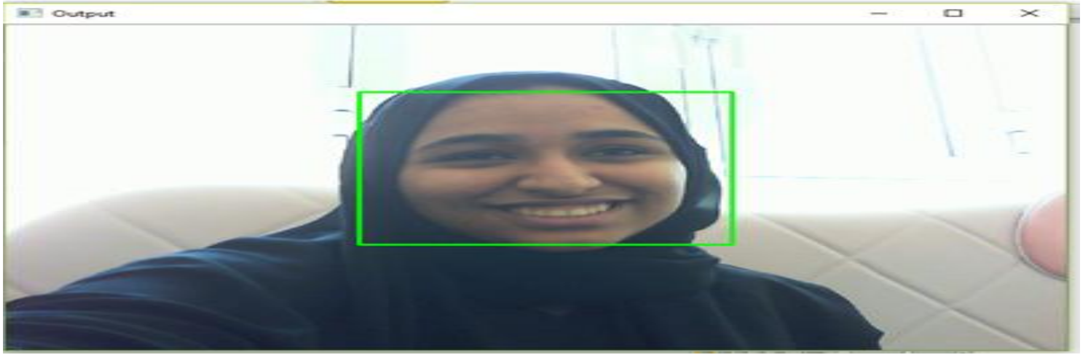


Figure 9. Real time face detected using camera HC 5000 in python 3.5

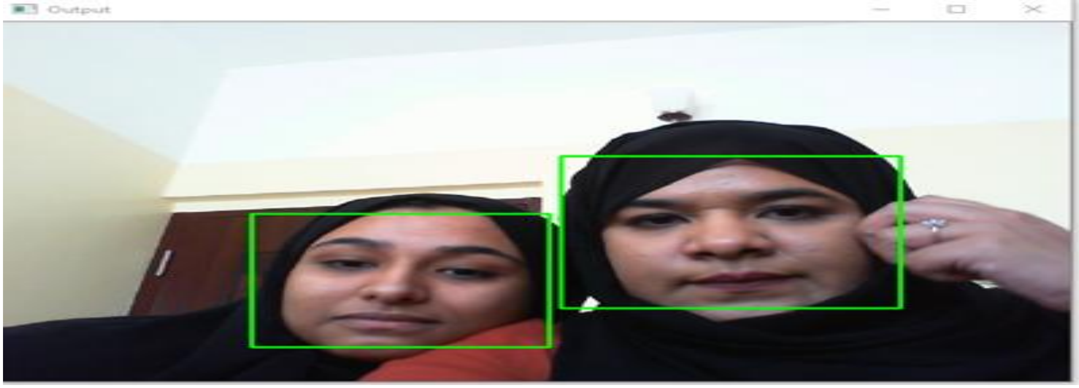
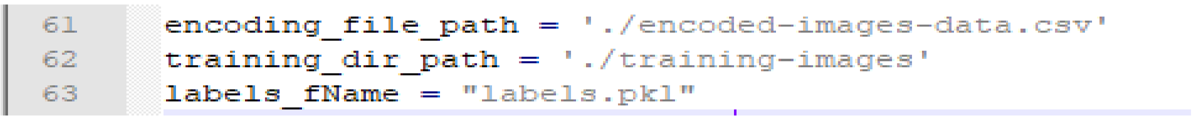


Figure 10. Multiple face detection using same algorithm in python 3.5

In second step face recognition is carried out with support of dataset built. The real time images are compared with trained model of dataset. The images are named if they are recognized on comparison with dataset built. In order to perform recognition images-data.csv and train.py is required. The images recognized are labelled using the training part as in figure 11.

The above files of database images are trained for the face recogntion model



```
encoding_file_path = './encoded-images-data.csv'
training_dir_path = './training-images'
labels_fName = "labels.pkl"
```

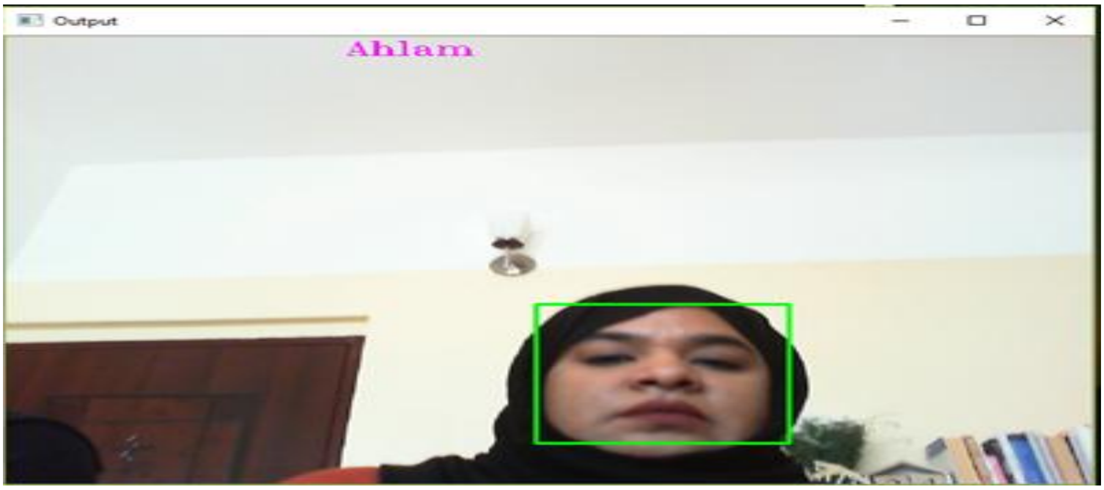


Figure 11. Shows the face recognized with the trained dataset and labelled with the name ‘Ahlam’ as well

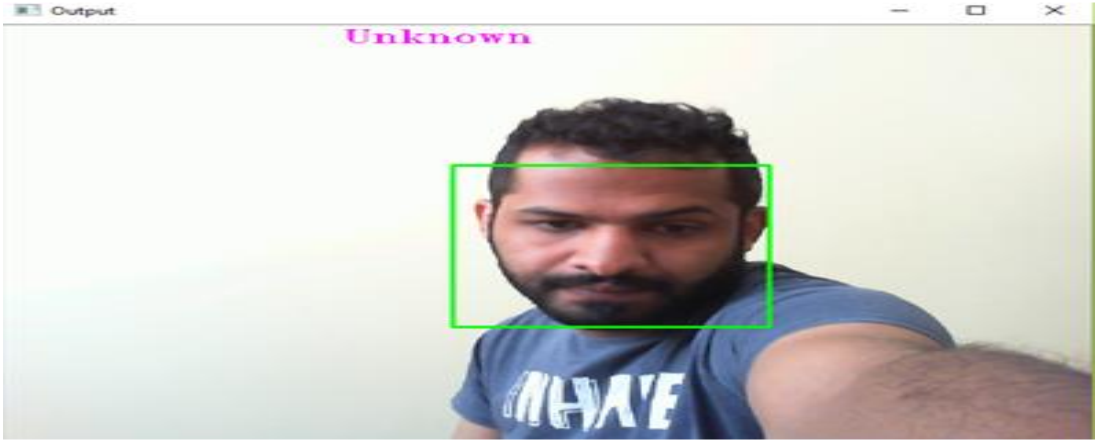
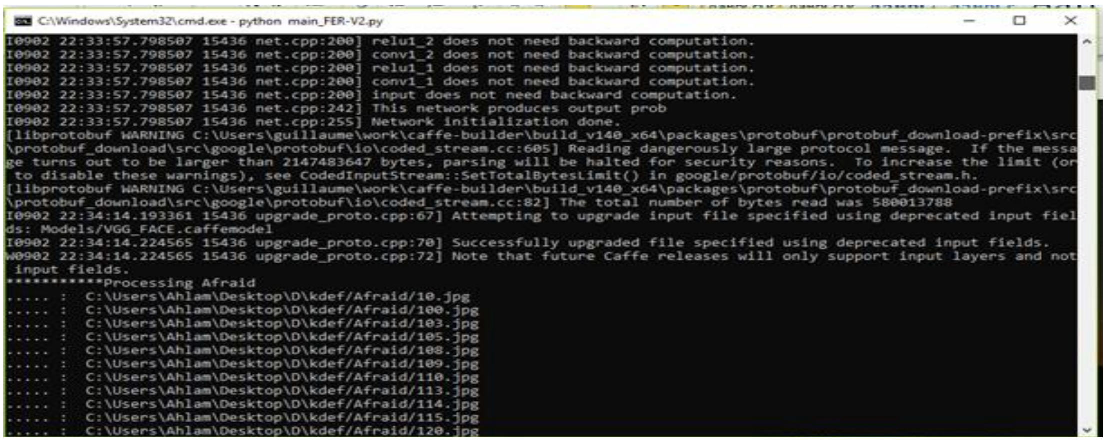


Figure 12. Shows the face unrecognized as unknown as the image is not trained with the built dataset for the model.

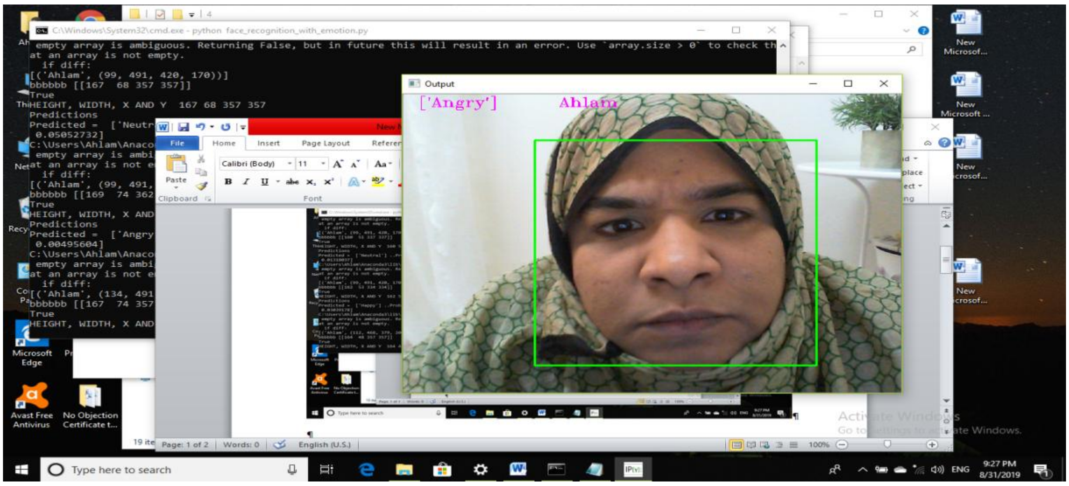
Third phase in the proposed work is face classification where the trained model is further classified to recognize the facial emotion and classify each expression with VGG 16 and svc_FER_linear_2.sav model. The face encodings are generated by convolution neural network (CNN) with knearest neighbour as clf. All the models are classified based on their weights to recognize the facial expression of human face in python main_FER_V2.py. Finally the program is run in different phases to load KDEF model to recognize the facial emotion expression as shown in the figure 12

The python file (face_recognition_with emotion.py programmed in Anaconda to classify the face emotion expressions.



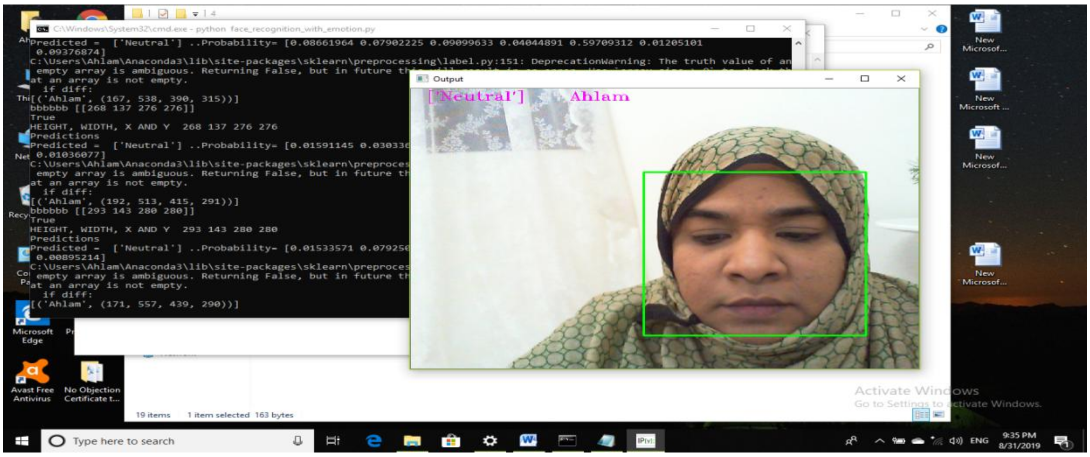
```
I0902 22:33:57.798507 15436 net.cpp:200] relu1_2 does not need backward computation.
I0902 22:33:57.798507 15436 net.cpp:200] conv1_2 does not need backward computation.
I0902 22:33:57.798507 15436 net.cpp:200] relu1_1 does not need backward computation.
I0902 22:33:57.798507 15436 net.cpp:200] conv1_1 does not need backward computation.
I0902 22:33:57.798507 15436 net.cpp:200] input does not need backward computation.
I0902 22:33:57.798507 15436 net.cpp:242] This network produces output prob
I0902 22:33:57.798507 15436 net.cpp:255] Network initialization done.
[libprotobuf WARNING C:\Users\guillaume\work\caffe-builder\build_v140_x64\packages\protobuf\protobuf_download-prefix\src
\protobuf_download\src\google\protobuf\io\coded_stream.cc:605] Reading dangerously large protocol message.  If the messa
ge turns out to be larger than 2147483647 bytes, parsing will be halted for security reasons.  To increase the limit (or
 to disable these warnings), see CodedInputStream::SetTotalBytesLimit() in google/protobuf/io/coded_stream.h.
[libprotobuf WARNING C:\Users\guillaume\work\caffe-builder\build_v140_x64\packages\protobuf\protobuf_download-prefix\src
\protobuf_download\src\google\protobuf\io\coded_stream.cc:82] The total number of bytes read was 580013788
I0902 22:34:14.193361 15436 upgrade_proto.cpp:67] Attempting to upgrade input file specified using deprecated input fiel
ds: Models/VGG_FACE.caffemodel
I0902 22:34:14.224565 15436 upgrade_proto.cpp:70] Successfully upgraded file specified using deprecated input fields.
W0902 22:34:14.224565 15436 upgrade_proto.cpp:72] Note that future Caffe releases will only support input layers and not
 input fields.
***********Processing Afraid
..... :  C:\Users\Ahlam\Desktop\D\kdef/Afraid/10.jpg
..... :  C:\Users\Ahlam\Desktop\D\kdef/Afraid/100.jpg
..... :  C:\Users\Ahlam\Desktop\D\kdef/Afraid/103.jpg
..... :  C:\Users\Ahlam\Desktop\D\kdef/Afraid/105.jpg
..... :  C:\Users\Ahlam\Desktop\D\kdef/Afraid/108.jpg
..... :  C:\Users\Ahlam\Desktop\D\kdef/Afraid/109.jpg
..... :  C:\Users\Ahlam\Desktop\D\kdef/Afraid/110.jpg
..... :  C:\Users\Ahlam\Desktop\D\kdef/Afraid/113.jpg
..... :  C:\Users\Ahlam\Desktop\D\kdef/Afraid/114.jpg
..... :  C:\Users\Ahlam\Desktop\D\kdef/Afraid/115.jpg
..... :  C:\Users\Ahlam\Desktop\D\kdef/Afraid/120.jpg
```

The facial emotion expressions are classified in the proposed work are angry, sad, happy, afraid, disgust and surprise like those shown in figures 13 to 18.


C:\Windows\System32\cmd.exe - python face_recognition_with_emotion.py
Output
['Angry']
Ahlam
Microsoft Edge
Avast Free Antivirus
No Objection Certificate t...
Type here to search
9:27 PM
8/31/2019
ENG

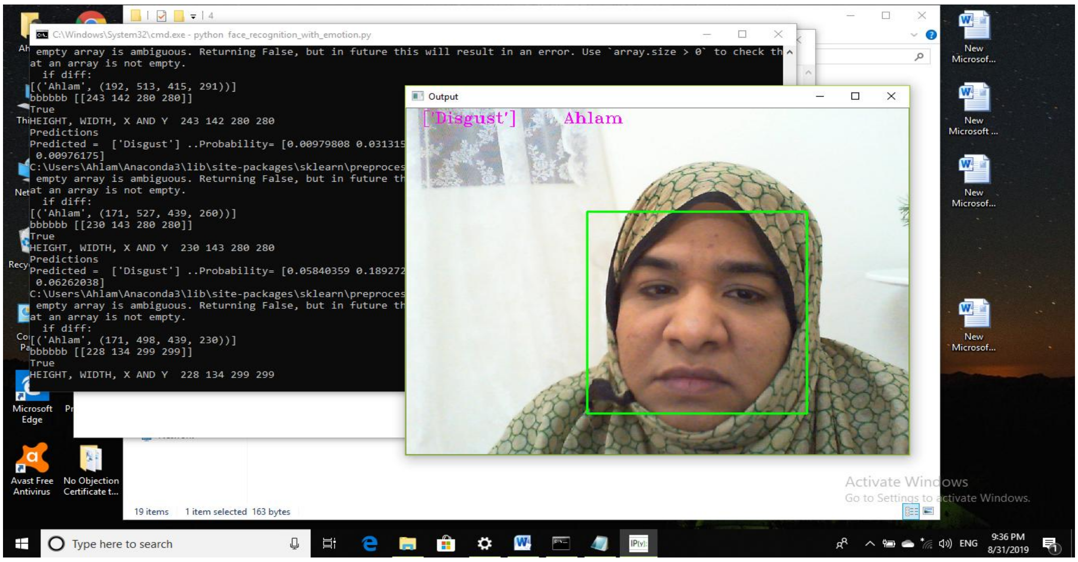
```
 empty array is ambiguous. Returning False, but in future this will result in an error. Use `array.size > 0` to check th
at an array is not empty.
  if diff:
[('Ahlam', (99, 491, 420, 170))]
bbbbbb [[167  68 357 357]]
True
HEIGHT, WIDTH, X AND Y  167 68 357 357
Predictions
Predicted =  ['Neutr
 0.05052732]
C:\Users\Ahlam\Anaco
 empty array is ambi
at an array is not e
  if diff:
[('Ahlam', (99, 491,
bbbbbb [[169  74 362
True
HEIGHT, WIDTH, X AND
Predictions
Predicted =  ['Angry
 0.00495604]
C:\Users\Ahlam\Anaco
 empty array is ambi
at an array is not e
  if diff:
[('Ahlam', (134, 491
bbbbbb [[167  74 357
True
HEIGHT, WIDTH, X AND
```

Figure 13. Face emotion classified from the emotion expression is **angry**


C:\Windows\System32\cmd.exe - python face_recognition_with_emotion.py
Output
['Neutral']
Ahlam
Microsoft Edge
Avast Free Antivirus
No Objection Certificate t...
19 items 1 item selected 163 bytes
Activate Windows
Go to Settings to activate Windows.
Type here to search
9:35 PM
8/31/2019
ENG

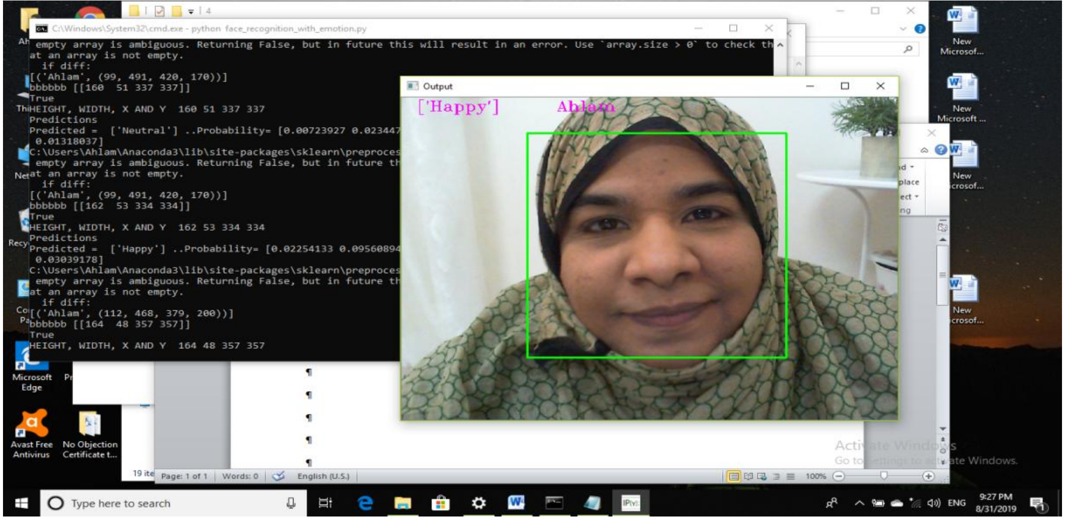
```
Predicted =  ['Neutral'] ..Probability= [0.08661964 0.07902225 0.09099633 0.04044891 0.59709312 0.01205101
 0.09376874]
C:\Users\Ahlam\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:151: DeprecationWarning: The truth value of an
 empty array is ambiguous. Returning False, but in future th
at an array is not empty.
  if diff:
[('Ahlam', (167, 538, 390, 315))]
bbbbbb [[268 137 276 276]]
True
HEIGHT, WIDTH, X AND Y  268 137 276 276
Predictions
Predicted =  ['Neutral'] ..Probability= [0.01591145 0.030336
 0.01036077]
C:\Users\Ahlam\Anaconda3\lib\site-packages\sklearn\preproces
 empty array is ambiguous. Returning False, but in future th
at an array is not empty.
  if diff:
[('Ahlam', (192, 513, 415, 291))]
bbbbbb [[293 143 280 280]]
True
HEIGHT, WIDTH, X AND Y  293 143 280 280
Predictions
Predicted =  ['Neutral'] ..Probability= [0.01533571 0.079250
 0.00895214]
C:\Users\Ahlam\Anaconda3\lib\site-packages\sklearn\preproces
 empty array is ambiguous. Returning False, but in future th
at an array is not empty.
  if diff:
[('Ahlam', (171, 557, 439, 290))]
```

Figure 14. The face expression classified from the image is neutral, the above figure shows the neutral state as partial or impartial way


C:\Windows\System32\cmd.exe - python face_recognition_with_emotion.py
Output
['Disgust'] Ahlam
Activate Windows
Go to Settings to activate Windows.
19 items 1 item selected 163 bytes
Type here to search
9:36 PM
8/31/2019

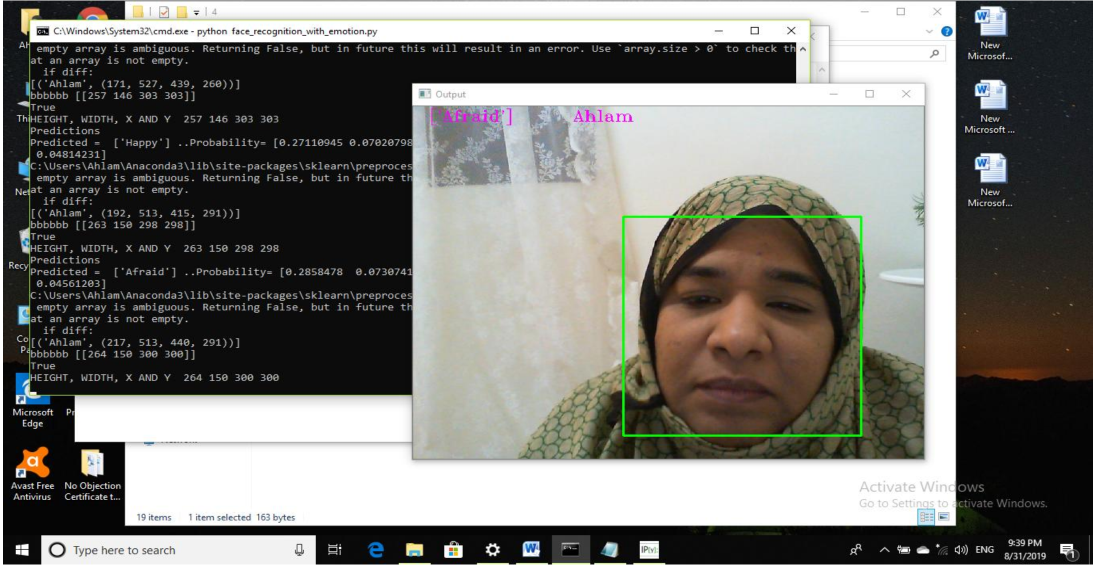
```
 empty array is ambiguous. Returning False, but in future this will result in an error. Use `array.size > 0` to check th
at an array is not empty.
  if diff:
[('Ahlam', (192, 513, 415, 291))]
bbbbbb [[243 142 280 280]]
True
HEIGHT, WIDTH, X AND Y  243 142 280 280
Predictions
Predicted =  ['Disgust'] ..Probability= [0.00979808 0.031315
 0.00976175]
C:\Users\Ahlam\Anaconda3\lib\site-packages\sklearn\preproces
 empty array is ambiguous. Returning False, but in future th
at an array is not empty.
  if diff:
[('Ahlam', (171, 527, 439, 260))]
bbbbbb [[230 143 280 280]]
True
HEIGHT, WIDTH, X AND Y  230 143 280 280
Predictions
Predicted =  ['Disgust'] ..Probability= [0.05840359 0.189272
 0.06262038]
C:\Users\Ahlam\Anaconda3\lib\site-packages\sklearn\preproces
 empty array is ambiguous. Returning False, but in future th
at an array is not empty.
  if diff:
[('Ahlam', (171, 498, 439, 230))]
bbbbbb [[228 134 299 299]]
True
HEIGHT, WIDTH, X AND Y  228 134 299 299
```

Figure 15. The face expression classified from the image is Disgust state which defines unsatisfaction level


C:\Windows\System32\cmd.exe - python face_recognition_with_emotion.py
Output
['Happy'] Ahlam
Page: 1 of 1 Words: 0 English (U.S.)
Type here to search
9:27 PM
8/31/2019

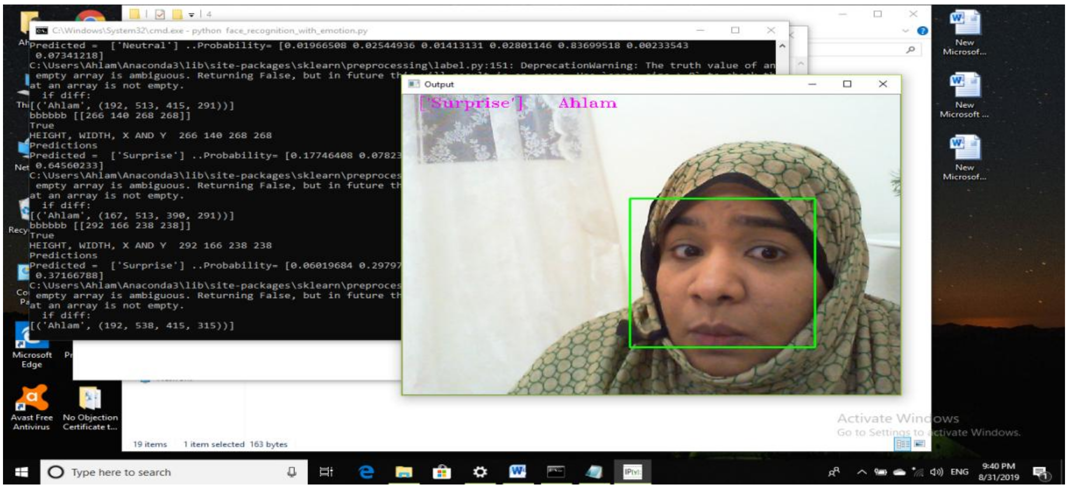
```
 empty array is ambiguous. Returning False, but in future this will result in an error. Use `array.size > 0` to check th
at an array is not empty.
  if diff:
[('Ahlam', (99, 491, 420, 170))]
bbbbbb [[160  51 337 337]]
True
HEIGHT, WIDTH, X AND Y  160 51 337 337
Predictions
Predicted =  ['Neutral'] ..Probability= [0.00723927 0.023447
 0.01318037]
C:\Users\Ahlam\Anaconda3\lib\site-packages\sklearn\preproces
 empty array is ambiguous. Returning False, but in future th
at an array is not empty.
  if diff:
[('Ahlam', (99, 491, 420, 170))]
bbbbbb [[162  53 334 334]]
True
HEIGHT, WIDTH, X AND Y  162 53 334 334
Predictions
Predicted =  ['Happy'] ..Probability= [0.02254133 0.09560894
 0.03039178]
C:\Users\Ahlam\Anaconda3\lib\site-packages\sklearn\preproces
 empty array is ambiguous. Returning False, but in future th
at an array is not empty.
  if diff:
[('Ahlam', (112, 468, 379, 200))]
bbbbbb [[164  48 357 357]]
True
HEIGHT, WIDTH, X AND Y  164 48 357 357
```

Figure 16. The face expression classified from the image is happy as satisfied


C:\Windows\System32\cmd.exe - python face_recognition_with_emotion.py
Output
['Afraid'] Ahlam
Activate Windows
Go to Settings to activate Windows.
19 items 1 item selected 163 bytes
Type here to search
9:39 PM
8/31/2019

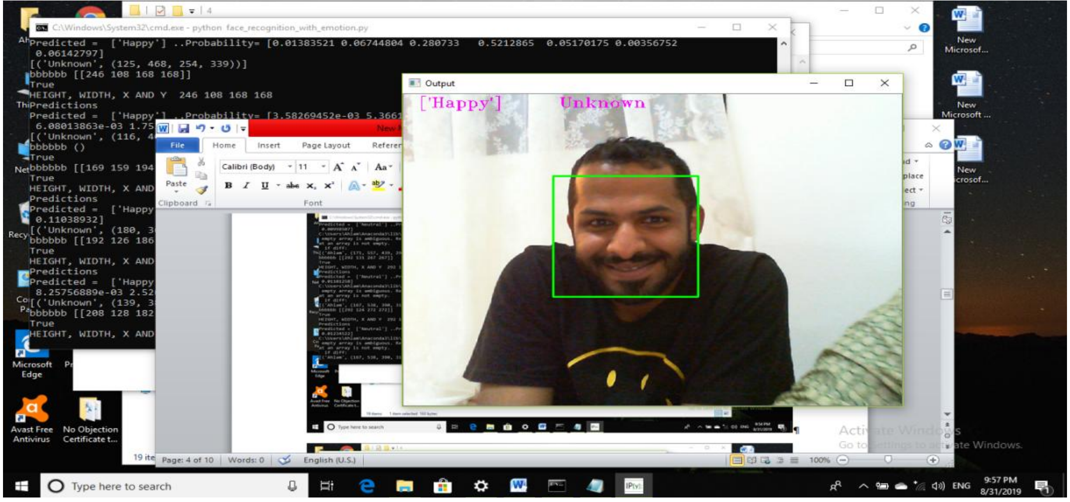
```
 empty array is ambiguous. Returning False, but in future this will result in an error. Use `array.size > 0` to check th
at an array is not empty.
  if diff:
[('Ahlam', (171, 527, 439, 260))]
bbbbbb [[257 146 303 303]]
True
HEIGHT, WIDTH, X AND Y  257 146 303 303
Predictions
Predicted =  ['Happy'] ..Probability= [0.27110945 0.07020798
 0.04814231]
C:\Users\Ahlam\Anaconda3\lib\site-packages\sklearn\preproces
 empty array is ambiguous. Returning False, but in future th
at an array is not empty.
  if diff:
[('Ahlam', (192, 513, 415, 291))]
bbbbbb [[263 150 298 298]]
True
HEIGHT, WIDTH, X AND Y  263 150 298 298
Predictions
Predicted =  ['Afraid'] ..Probability= [0.2858478  0.0730741
 0.04561203]
C:\Users\Ahlam\Anaconda3\lib\site-packages\sklearn\preproces
 empty array is ambiguous. Returning False, but in future th
at an array is not empty.
  if diff:
[('Ahlam', (217, 513, 440, 291))]
bbbbbb [[264 150 300 300]]
True
HEIGHT, WIDTH, X AND Y  264 150 300 300
```

Figure 17. The face expression classified from the image is Afraid



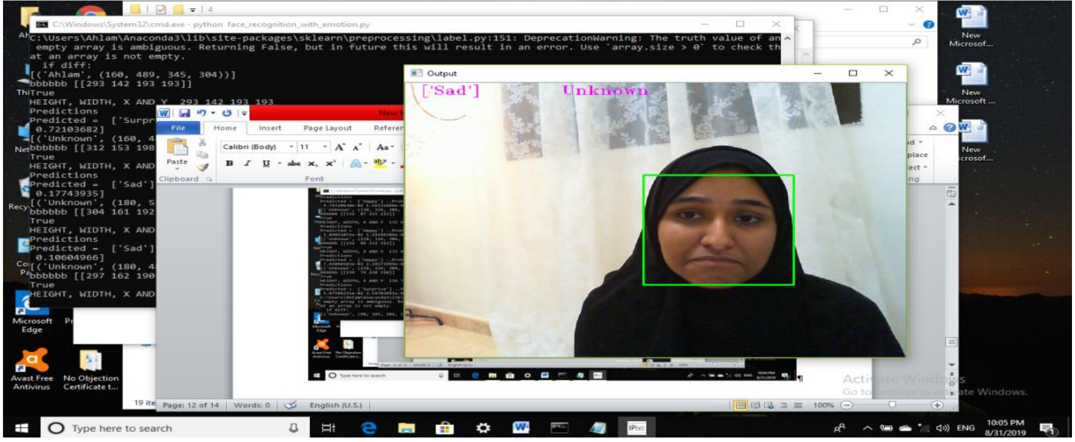
```
C:\Windows\System32\cmd.exe - python  face_recognition_with_emotion.py
Predicted =  ['Neutral'] ..Probability= [0.01966508 0.02544936 0.01413131 0.02801146 0.83699518 0.00233543
 0.07341218]
C:\Users\Ahlam\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:151: DeprecationWarning: The truth value of an
 empty array is ambiguous. Returning False, but in future th
at an array is not empty.
  if diff:
[('Ahlam', (192, 513, 415, 291))]
bbbbbb [[266 140 268 268]]
True
HEIGHT, WIDTH, X AND Y  266 140 268 268
Predictions
Predicted =  ['Surprise'] ..Probability= [0.17746408 0.07823
 0.64560233]
C:\Users\Ahlam\Anaconda3\lib\site-packages\sklearn\preproces
 empty array is ambiguous. Returning False, but in future th
at an array is not empty.
  if diff:
[('Ahlam', (167, 513, 390, 291))]
bbbbbb [[292 166 238 238]]
True
HEIGHT, WIDTH, X AND Y  292 166 238 238
Predictions
Predicted =  ['Surprise'] ..Probability= [0.06019684 0.29797
 0.37166788]
C:\Users\Ahlam\Anaconda3\lib\site-packages\sklearn\preproces
 empty array is ambiguous. Returning False, but in future th
at an array is not empty.
  if diff:
[('Ahlam', (192, 538, 415, 315))]
```

Figure 18. The face expression classified from the image is Surprise representing astonishing events



```
C:\Windows\System32\cmd.exe - python  face_recognition_with_emotion.py
Predicted =  ['Happy'] ..Probability= [0.01383521 0.06744804 0.280733   0.5212865  0.05170175 0.00356752
 0.06142797]
[('Unknown', (125, 468, 254, 339))]
bbbbbb [[246 108 168 168]]
True
HEIGHT, WIDTH, X AND Y  246 108 168 168
Predictions
Predicted =  ['Happy'] ..Probability= [3.58269452e-03 5.3661
 6.08013863e-03 1.75
[('Unknown', (116, 4
bbbbbb ()
True
bbbbbb [[169 159 194
True
HEIGHT, WIDTH, X AND
Predictions
Predicted =  ['Happy
 0.11038932]
[('Unknown', (180, 3
bbbbbb [[192 126 186
True
HEIGHT, WIDTH, X AND
Predictions
Predicted =  ['Happy
 8.25756889e-03 2.52
[('Unknown', (139, 3
bbbbbb [[208 128 182
True
HEIGHT, WIDTH, X AND
```

Figure 19. The face classification has been tested for untrained dataset image where expression are classified from the image is happy



```
C:\Windows\System32\cmd.exe - python  face_recognition_with_emotion.py
C:\Users\Ahlam\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:151: DeprecationWarning: The truth value of an
 empty array is ambiguous. Returning False, but in future this will result in an error. Use `array.size > 0` to check th
at an array is not empty.
  if diff:
[('Ahlam', (160, 489, 345, 304))]
bbbbbb [[293 142 193 193]]
True
HEIGHT, WIDTH, X AND Y  293 142 193 193
Predictions
Predicted =  ['Surpr
 0.72103682]
[('Unknown', (160, 4
bbbbbb [[312 153 198
True
HEIGHT, WIDTH, X AND
Predictions
Predicted =  ['Sad']
 0.17743935]
[('Unknown', (180, 5
bbbbbb [[304 161 192
True
HEIGHT, WIDTH, X AND
Predictions
Predicted =  ['Sad']
 0.10604966]
[('Unknown', (180, 4
bbbbbb [[297 162 190
True
HEIGHT, WIDTH, X AND
```

Figure 20. The face classification has been tested for untrained dataset image where expression are classified from the image as sad

However from the face classifcation it has been tested that different set of images as shown above in figures 19 to 21 are also identified using face classifier. The face expression is detected from unknown dataset images also but not recognized as dataset is not trained.

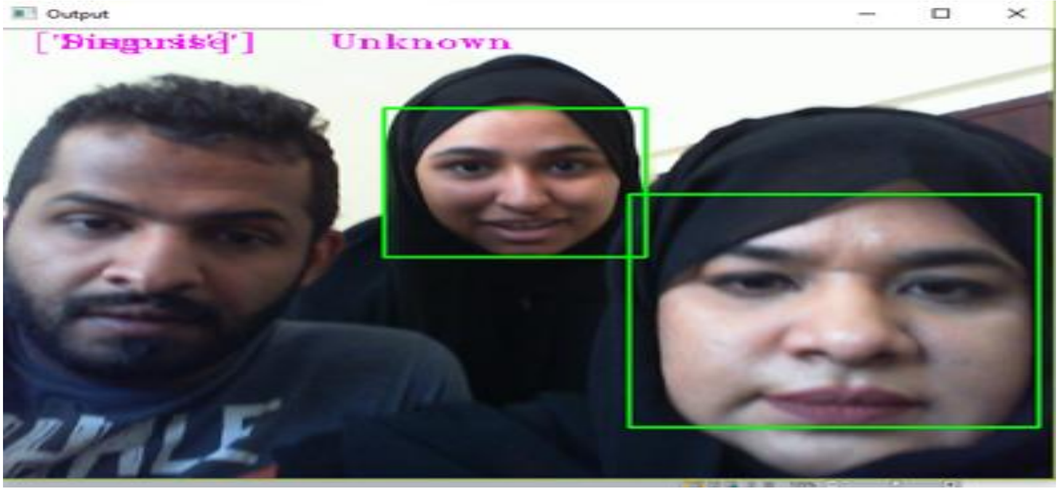


Figure 21. Multiple faces are also classified with facial expression but not recognized as there faces have not been trained with the dataset used. The face emotion classified are overwritten as angry, surprise and neutral as displayed in the picture.

Table 1. shows the validation accuracy with the dataset and VGG 16 CNN model used in testing and classification.

Model	Dataset	Accuracy/F1-score
VGG-16 Face	KDEF	0.88

The VGG 16 and KDEF dataset is used to classify the face with seven expressions and accuracy shown in table 1 to prove the performance of CNN model used.

5. Conclusion

The proposed work is designed to develop a real time system to detect, recognize and classify human face. The classified expressions are represented in seven states as shown in above results. The softwares used to test the functionality are Anaconda and python 3.5. For Face detection viola jones and haar cascade algorithm was used. Similarly KDEF dataset and VGG 16 were used with convolution neural network model for face recognition and classification. The dlib and other libraries works in support with python programming. The performance measures are validated with the CNN model designed with an accuracy of 88%. However the results prove that the network architecture designed has better advancements than existing algorithms. This application is widely used in many areas such as education, industry, medical and electronics etc. VGG 16 and KDEF dataset is built to achieve face classification and recognition. The above seven facial expressions shows different conditions of a person. Some of the applications direclty related are Autism which can be helpful to interpret expressions of a person or child. The emotions of a student E-learning techniques could also be evaluated easily through the proposed work.

Acknowledgments

The research work was supported by Middle East College, Muscat in providing the lab facilities and also creating an academic environment to achieve the desired results. We are also thankful to all the esteemed professors in the department who moderated the manuscript at each step. We are immensely grateful to Prof. Asif Hussain shaik to define new research findings related to the work proposed.

References

[1] Ken Nozaki, HisaoIshibuchi 1996. Adaptive fuzzy rule based Classification systems, IEEE transactions on Fuzzy systems, Vol.**4**, No.3.

[2] Farid Ghareh Mohammadi, Mohammed Saniee Abadeh 2014. Image Steganalysis using abee colony based feature selection Algorithm. Engineering Applications of AI. Science Direct, pp. 35-43.

[3] Liu, F L C Y, 2015. Improving Steganalysis by Fusing SVM Classifiers for JPEG Images. IEEE, pp. 185-190.

[4] Chhikara, M K, 2016. Significance of feature selection for image Steganalysis.IEEE.

[5] Bin Li.2011 A survey on Image steganography and Steganalysis,Journal of Information Hiding and Multimedia signal Processing, Ubiquitous International, vol**2** No.3 pp. 142-172.

[6] Cho, B H, 2013. Block-based image Steganalysis: Algorithm and performance evaluation. Elsevier, 24(7), pp. 846-856.

[7] Alex Krizhevsky, I. S. a. G. E. H., 2012. Imagenet classification with deep convolutional neural networks. ACM DL, Volume 1, p. 1097–1105.

[8] C. Shan, S. G. a. P. W. M., 2009. “Facial expression recognition based on local binary patterns: A comprehensive study. Image and Vision Computing, 27(6), p. 803–816.

[9] Dharma, A. S. A. a. D., April 2018. Novel directional patterns and a Generalized Supervised Dimension Reduction System (GSDRS) for facial emotion recognition. Springer Link, 77(8), p. 9455–9488 .

[10] Hao Wang, S. W. a. B. F., August 2018. Facial expression recognition using iterative fusion of MO-HOG and deep features. The Journal of Supercomputing.

[11] Ji, Y. Z. a. Q., May 2005. Active and Dynamic Information Fusion for Facial Expression Understanding from Image Sequences. IEEE, 27(5), p. 699–714.

[12] Mahdi Jampour, T. M. a. H. B., 2015. Multi-view Facial Expressions Recognition using Local Linear Regression of Sparse Codes. Seggau, Austria, Research Gtae

[13] Mundher Al-Shabi, W. P. C. a. T. C., Aug 2019. Facial Expression Recognition Using a Hybrid CNN– SIFT Aggregator. ResearchGate(RG), pp. 139-149.

[14] Viraj Mavani, S. R. a. K. P. M., 2017. Facial Expression Recognition using Visual Saliency and Deep Learning. s.l., IEEE.

[15] CUDA Toolkit 10.1 Update 2 available from <https://developer.nvidia.com/cuda-downloads> [2 September 2019]

[16] KDEF | The Kenya Dryland Education Fund (2019) available from <https://kdef.org/> [2 September 2019]