Facial Expression Detection and Recognition using VGG- 16

Aradhana Singh Parihar
M.Tech. Scholar SAGE University, Indore

Shweta Agrawal, PhD
Professor, CSE SAGE University, Indore

ABSTRACT
Facial expression recognition software is useful in a variety of circumstances. In recent years, there has been a lot of research on facial expression detection and recognition. Facial expression recognition software is useful in a variety of circumstances such as, security, camera surveillance, criminal investigations, smart card applications, database management systems, and in modern devices for identity verification etc. This paper shows how to implement facial expression detection and recognition system. The facial recognition is to recognize and validate facial traits. However, Haar cascade detection is used to captured facial features in real time. In three different phases the sequential process work can be define, In the first step, a camera detects a human face, and the acquired input is processed based on features with the help of the Keras convolutional neural network model database. Human faces are validated in the third stage to categorize human emotions as happy, neutral, furious, sad, and surprised. This suggested study is broken down into two aspirations: face detection and expression identification. This work will comes under computer vision field. We will use opencv, keras and python programming in this project. The testing result illustrates the system's perfection in detecting and recognizing facial expressions. Finally, we will be able to obtain accurate facial expression detection and identification results.

Keywords
Facial Expression Detection, VGG- 16

1. INTRODUCTION
In the past, computer vision problems were difficult to solve, but with the advancement of modern technologies, problems such as varying light, age, hair, and other accessories are no longer a problem. Face recognition software, on the other hand, is used to make it easier to identify and authenticate people based on their facial features [1][2]. As a result, evaluating facial features as well as changes in their expression and movements is essential. These characteristics and expressions aid in the classification of human facial emotions [3][4]. To recognize and classify human faces, various approaches are necessary,

Machine learning [5][6] is a sub domain of artificial intelligence and has many algorithm that can be used for face recognition. For handling huge data and getting better results deep learning [7][8] can be used. Although there are many challenges exist in succesfull implementation of these algorithms[9][10], then also there exist applications of deep learning in medical domain, stock market, agriculture , and many more domain [11-13]. Preprocessing, detection, orientation, feature extraction, and emotion classification are all steps in the facial expression identification and classification process[14-18]. This learning usually takes the form of a neural network model, in which neurons serve as inputs and are coupled to move as outputs. Similar to machine learning, deep learning algorithms are interconnected, but there are multiple levels of these algorithms, each of which interprets the data they introduce differently. Because their action is a source of inspiration, or we may say; an attempt to emulate the role of the human neural networks in the brain, this network of algorithms is known as the network of artificial neurons. Deep neural networks can examine data functions in the so-called functional hierarchy thanks to several hidden levels, which allow simple functions, such as two pixels, to integrate from one level to the next, generating, for example, more complicated functions. Low-level networks are less capable of processing mathematical operations than multilevel networks, which can process large amounts of data . Implementation process is divided into three aspects: facial expression detection, recognition, and emotion classification.

Computer vision is the study of how computers can see and recognize digital images and videos in the same way that humans do. The difficulties that computer vision encounters are partly due to our lack of understanding of biological vision.
The process of recording, processing, analyzing, and comprehending digital images in order to extract high-dimensional data from the real world and generate symbolic or numerical data that may be used to make decisions is known as computer vision. The techniques employed in this procedure include object detection, video tracking, motion estimation, and image restoration.

Convolution Neural Networks
In recent years, the development of Convolution neural networks has had a substantial impact on the computer vision domain, as well as a crucial step and capacity to recognize objects or real-world objects. Because of the increasing value of the computer and the amount of data available to create a neural network, the technique has gained in popularity.

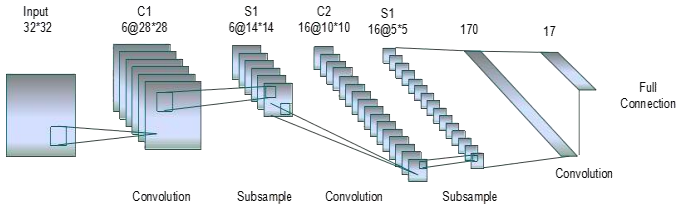


Figure 1. Convolution Neural Networks

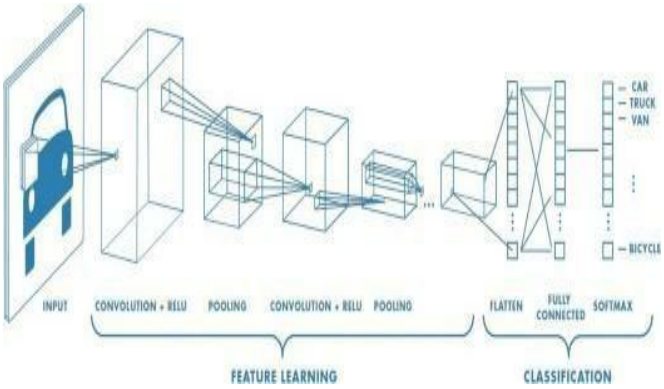


Figure 2. Main process of Convolution Neural Networks

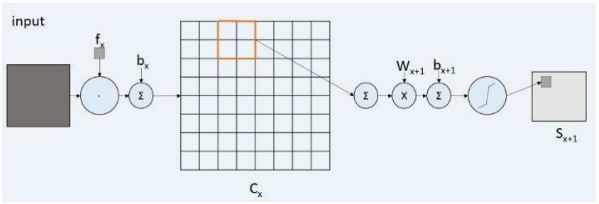


Figure 3..stages of Convolution neural network (CNN) model

Kernels are used to detect boundary functions or outline for an image in convolutional neural network (CNN) models. Weights are organized in an array of values in this model to form and get desired qualities. Every CNN model allots space to determine the picture control to be identified. The values in the image show the degree to which the convolution operation is dependent on the location, and so the product is computed and determined with the location.

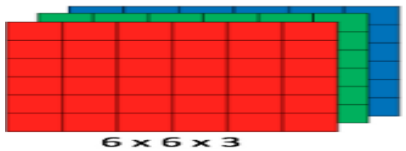


Figure 4. Array of RGB Matrix

2. APPROACH

Face Detection, Face Recognition, and Face Classification are three consecutive processes in the proposed work. The first stage involves using a video camera to capture a human face and determining its exact location using bounding box coordinates for the face recognized in real-time. Face detection is performed using Haar cascade detection with the open CV library in this step.To detect human faces, the Viola Jones algorithm and haar cascade characteristics are integrated[19]. Shapes, objects, and landscapes are among the images recognized. Human faces are recognized in this phase, and face features are retrieved and stored in a database for face recognition. In order to process face detection, identification, and classification, the distribution platform employs Anaconda and Python 3.5 software.The image can be found in the database and other libraries. Face detection is done first, followed by database feature recognition and matching utilizing the CNN model training and testing database. Finally, the recognized human face is categorized as angry, ,sad, joyful, neutral, or surprise based on the expression in real time.For massive database recognition and classification, the VGG 16 network design uses the CNN paradigm. For detecting human faces, the local binary model histogram is employed as an open CV library. To accept test data from the machine, the circular binary model is constructed to the indicated areas with appropriate labels.Create a folder and name it, then write two python files to do the procedural outcome of face detection. Set and run complete files from the Integrated Development Environment of Python, including classification little vgg.py, facial

expression recognization .py, and the haarcascade frontal face default.xml library.

Proposed Functional Design

First of all make a folder and give name , then code two python files for procedural outcome of face detection. Set and run complete files from the Integrated Development Environment of Python, including create data.py, face recognize.py, and the haarcascade frontal face default.xml library. To construct and train the dataset for identification and emotion classification, the collected image is modelled using network architecture[25]. The information is presented as a flow chart in which the processes are given in order.

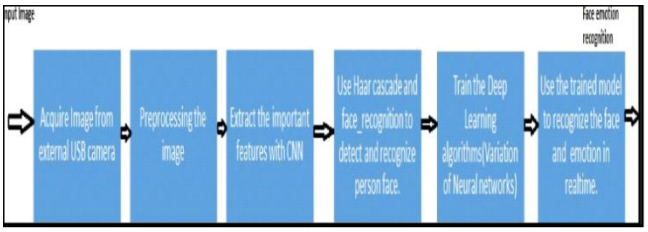


Figure 5.Block diagram

Process in the form of Flowchart and Design Specifications

Figure 5 shows the real-time face detection, recognition, and classification.The acquired image is bounded in a box, transformed to a binary pattern, and saved in a database as a feature vector. The images are given training to replicate the input image as well as characterise facial expressions as neutral, happy, angry, sad ,and surprise. seven steps in the training process: importing the dataset, preprocessing the data, supplementing the data as a feature vector, developing and compiling the design model, training and storing the feature vector, and testing the test model.

3. IMPLEMENTATION

We must first obtain a dataset from Google. This dataset will be used to train the machine learning algorithm. The dataset is divided into two folders, one for training and the other for validation. Each folder contains photos of emotions such as angry, sad, happy, natural, and surprise, resulting in a total of five classes. To import dense, dropout, activation, flatten, and batch normalization, we'll utilisekeras. layers.

Creating a database that includes an open CV library as well as other resources

a) Create a folder and name it, then create two python files, facial expression recognition.py and classification little vgg. py. To check for problems, the code is copied into the resulting source file and executed. To support the facial features, copy the xml file into the project directory. Haarcascadefrontalface default.xml is the file that has to be duplicated.

c) Keras: Keras is a Python high-level library that runs on top of the Tensorflow framework. It was created with the goal of better comprehending deep learning approaches, such as layering neural networks while keeping shape notions and mathematical intricacies.

The types of framework creation are as follows:

- Sequential API
- Functional API

To develop a deep learning model in Keras, we took the following 8 steps into consideration:

- Loading the data
- Preprocess the loaded data
- Definition of model
- Compiling the model
- Fit the specified model
- Evaluate it
- Make the required prediction
- Save the model

FROM KERAS.MODELS-

Sequential api-the sequential api here used to make model layer by layer.

Dense class- The regular deeply linked neural network layer is the dense class -dense layer. The following operation is performed on the input by the dense layer, and the output is returned.

output=activation(dot(input,kernel)+bias)

where input is the input data, kernel denotes the weight data, dat denotes the numpy dot product of all input and its matching weights, and bias denotes a bias value used in machine learning to improve accuracy.

Dropout class-The dropout layer, which helps prevent overfitting, sets input units to 0 at a frequency of rate at each step throughout training time. Inputs that are not set to 0 are scaled up by 1/(1-rate) so that the total sum of all inputs .

BatchNormalization -

batch normalisation is a modification that keeps the mean output around 0 and the standard deviation output near 1. Importantly, batch normalisation behaves differently during inference and training. The layer normalises its output during training by using the mean and standard deviation of the current batch of inputs.

The layer normalises its outputs during inference by taking a moving average of the mean and standard deviation of the batches it saw during training.

Conv2d-The conv2d parameter specifies how many filters convolutional layers will learn from. It is an integer variable that also controls the number of convolution output filters.

Maxpooling2d- downsamples the input along its spatial dimensions (height and width) by taking the maximum value for each channel of the input over an input window (of size

determined by pool size). Each dimension of the window is adjusted by strides.

FROM KERAS.OPTIMIZERS-

RMSprop - The goal of RMSprop is to:
a)maintain a moving (discounted) average of squar of gradients
b)divide the gradient by the root of this average
This implementation of RMSprop uses plain momentum, not nesterov momentum.The centered version additionally maintains a moving average of the gradients, and uses that average to estimate the variance.

SGD- keras provides the sgd class that implements the stochastic gradient descent optimizer with a learning rate and momentum.First,an instance of the class must be created and configured,then specified to the optimizer argument when calling the fit() function on the model..

Adam-adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first -order and second -order moments.

FROM KERAS.CALLBACKS-

Modelcheckpoint- callback to save the keras model or model weight at some frequency.
modelcheckpoint callback is used in conjunction with training using model.fit() to save a model or weights (in a checkpoint file) at some interval,so the model or weight can be loaded later to continue the training can be loaded later to continue the trainingbfrom th state saved.

Earlystopping- stop training when a monitored metric has stopped improving. Assuming the goal of training is to minimize the loss.With this, the metric to be monitored would be 'loss', and model would be in'.A model.fit() traing loop will check at end of every epoch whether the loss is no longer decreasing, considering the min delta and patience if applicable. Once its found no longer decreasing,model.stop training is marked true and the training terminates.
The quantity to be monitored needs to be available in logs dict.To make it so,pass the loss or metrics at model.compile.

ReduceLROnPlateau- reduce learning rate when a metric has stopped improving.Models often benefit from reducing the learning rate by a factor of 2-10 once learning stagnates.This callback monitors a quantity the if no improvement is seen for a patient number of epoch, learning rate is reduced.

c)VGG model: This model is used to support and train Convolution neural networks as solvers and models, as well as data preparation and training. During the creation of Python scripts, this model is used to anticipate the unneeded data. The CNN Caffe model in Python programming is shown here.

d)Numpy: numpy is a python library used for working with arrays.It also has functions for working in domain of linear algebra,fourier transform, and matrices.Numpy was created in 2005 by travis oliphant.It is an open source project and you can use it freely.Numpy stands for numerical python.Numpy aims to provide an array object that is up to 50x faster than traditional python lists. The array object in numpy is called ndarray,it provides a lot of supporting functions that make working with ndarray very easy.

e)OpenCV: This is an open source computer vision library. This library will be used to do image transformations. The implementation of algorithms in open cv is diverse.cv2.imread()method loads and image from the specified file. If the image cannot be read (because of missing file, improper permissions,unsupported or invalid format) then this method returns and empty matrix.

f) Python : python is the most powerful programming language, and it is particularly good for handling statistical problems involving machine learning techniques and deep learning.

Training procedure for recognition and classification

The training and testing data is separated into two models, which are loaded using keras.models. The image utilised is 48x48 pixels, which is rescaled and retrieved as a feature vector before being used to train the classifier. The following is the code for extracting features for classifier training.

Creating all the files to Python 3.8 and Anaconda 4.8.2 software platform.

Code written using vscode editor in python language. There is two file were created for the whole project the first one is for training the data and the last one is for testing the output or get output.
First we did training, where From Keras.models we have imported sequential,dense,dropout,activation,flatten,batch normalization,conv2d,maxpooling2d.From keras.optimizers we have imported RMSprop,SGD,adam.
From keras.callbacks we have imported modelcheckpoint,earlystooping,ReducelronPlateau.
it takes 25 epochs to to generate emotion little.vgg file which will used in testing .

In the second step we did testing, code is written in file Facial expression recognition.py.here we use keras.processing.image to import image to array ,opencv is used to read images numpy is also used.here we use haarcascade and that file which ids generated when we train the data that is emotion little.vgg .

After executing testing part that is done by running Facial expression recognition.py we will able to see the desired output. The software ppllatform used in prorocessing the results is Python 3.8 and aaAnaconda 4.8.2.

4. FIGURES USED IN IMPLEMENTATION

Training

Number of classes =5
Image size= 48,48
Batch size (how many images want to train at once)=8

Values for image data generator for train datagen
Rescale image =1/255
Rotation range=30
Shear range=0.3
Zoom range=0.3
Width shift range=0.4
Height shift range=0.4
Horizontal flip=true

Vertical flip =true
Fill mode ='nearest '"
Values for image data generator for validation datagen
Rescale =1/255

For train generator and validation generators
"Color mode ='grayscale'
Target size=(48,48)
Batch size=8
Class mode ='categorical'
shuffle=true"
Sequential model

Model 1:

Conv2d (32, (3,3),padding ="same',kernel initializer='he normal',input shape=(48,48,1))
Activation ('elu')
BatchNormalization ()
Maxpooling2d(pool size =(2,2))
Dropout (0.2)

"Conv2d (64, (3,3),padding ="same',kernel initializer='he normal')
Activation ('elu')
BatchNormalization ()
Maxpooling2d(pool size =(2,2))
Dropout (0.2)"[22]

Conv2d (128, (3,3),padding ="same',kernel initializer='he normal')
Activation ('elu')
BatchNormalization ()
Maxpooling2d(pool size =(2,2))
Dropout (0.2)

"Conv2d (256, (3,3),padding ="same',kernel initializer='he normal')
Activation ('elu')
BatchNormalization ()
Maxpooling2d(pool size =(2,2))
Dropout (0.2)

"Flatten ()
Dense (64,kernel initializer ='he normal ')
Activation ('elu')
BatchNormalization ()
Dropout (0.5)

Dense (64,kernel initializer ='he normal ')
Activation ('elu')
BatchNormalization ()
Dropout (0.5)

Dense (num classes, kernel initializer ='he normal')
Activation ('softmax')"

Testing

"For capturing video =cv2. videocapture (0)# i use 0 because my camera is internal camera .for external camera 1 will be used.

"Grab a single frame of video
Gray=cv2.cvtcolor (frame,cv2.COLOR BG2 GRAY)
Faces=face classifier. Detectmultiscale (gray,1,3,5)

Rectangle (frame (x,y), (x+w,y+h), (255,0,0),2)

Resize (rio gray, (48,48),interpolation =cv2.Inter area)

Rectangle,face,image= face detector (frame)
Astype ('float')/255.0
Expand dims (rio,axis=0)

Make a prediction on the roi, then lookup the class
Predict (rio)(0)"[21]

PutText (frame,label,label position, cv2.FONT HERSHEY SIMPLEX,2, (0,255,0),3)

for no image found
PutText (frame,'no face found',(20,60), cv2.FONT HERSHEY SIMPLEX,2, (0,255,0),3)

Imshow ('emotion detector',frame)
Waitkey (1)

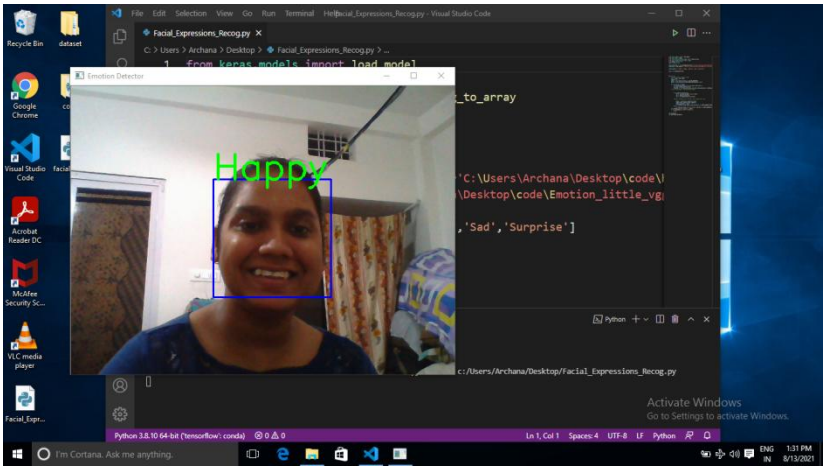
5 . RESULT

We have made 2 files of code where 1st file that is classification little. Py contains code for training data and the another one that is facial expression recog.py contains code for using trained data and display recognition part .

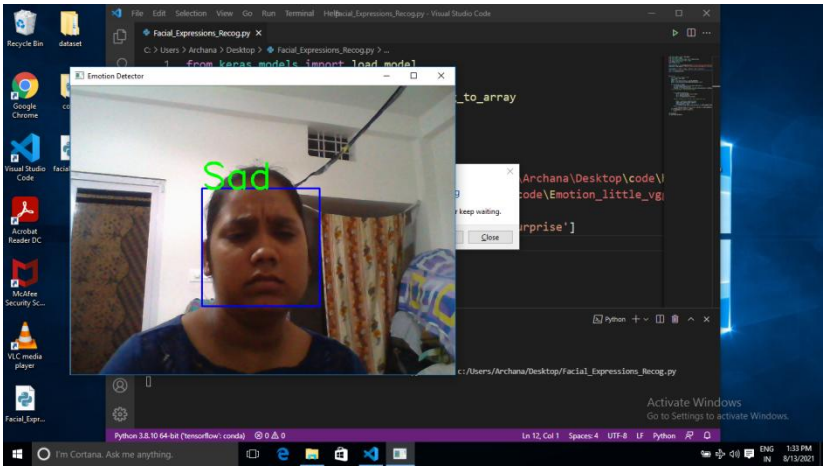
*The first one will executed first , then it will start training data , where it will took 25 epoch to get completed . Once training part get completed we will get one file of training data that is emotion little.vgg .

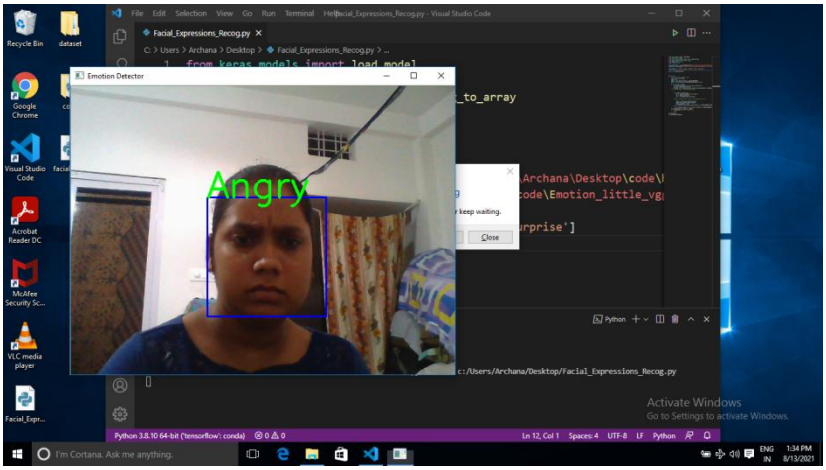
*Now after doing this we will go to second file that is facial expression recog.py here we use emottionlittle.vgg file and a haar cascade frontal face file to get our required output .Along with all external file we use multiple libraries of python which are used to do different function .

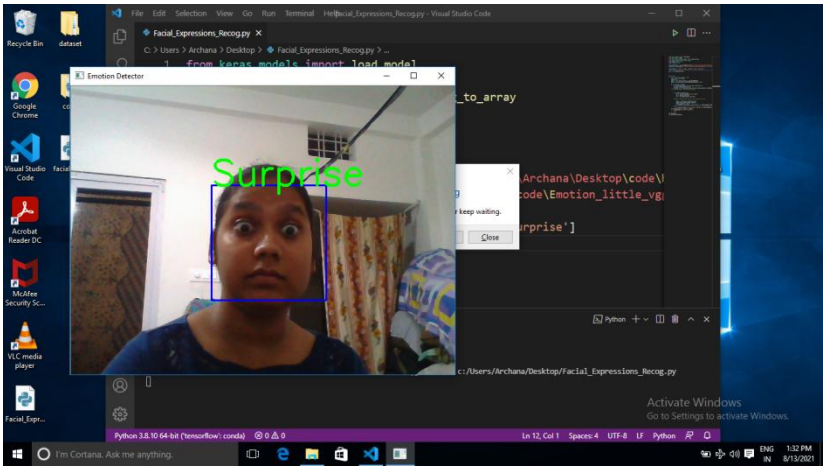
*At the last we will going to check our output so we have run facial expression recog .py file .once we run it first it will start the camera or external camera of computer .Now we have to sit in front of came and give expressions .the system will recognize the expression and display at the top of the border.

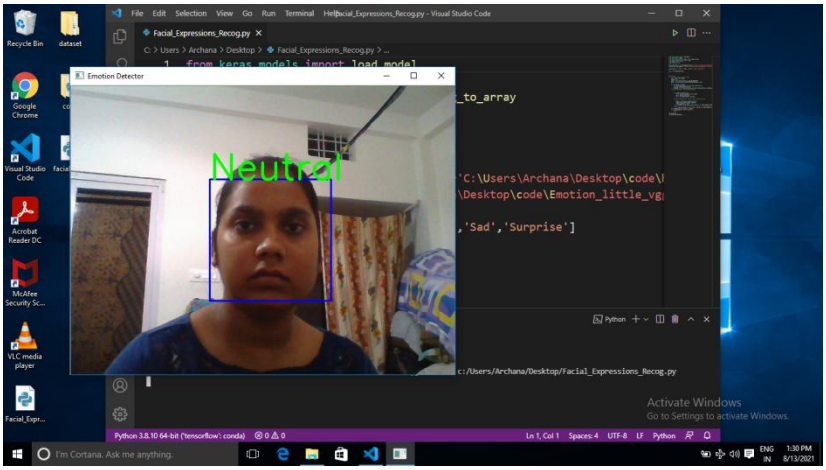


The first expression here is happy in front of Computer .you can clearly see that system recognized the expression by displaying that the face is coverd in squar with color blue and emotion name is displayed at the top of the squar border









Similarly all the expression were tested for confirmation .

6. CONCLUSION

The goal of this project is to create a real-time system that can detect, recognise, and classify human facial expressions. As indicated in the above data, the categorised expressions are represented in five states. Anaconda and Python 3.8 are the softwares that were utilised to test the functionality.The Haar cascade method was used to detect faces. Similarly, for face recognition and classification, VGG 16 is combined with a convolutional neural network model. Python programming is aided by the keras and other libraries. The CNN model, which has an accuracy of 88 percent, is used to validate the performance measurements.The results, on the other hand, show that the network architecture designed outperforms previous techniques. This programme is widely utilised in a variety of fields, including medicine, manufacturing, education, and electronics. Face categorization and identification are achieved using VGG 16 and the dataset.The five facial expressions shown above depict various aspects of a person's condition. Autism is one of the applications that is directly associated and can be used to read a person's or child's expressions. The proposed study could also be used to assess a student's emotions when using E-learning strategies.

7. REFERENCES

[1] B. Fasel and J. Luettin, "Automatic facial expression analysis: a survey," Pattern Recognition, vol. 36, no. 1, pp. 259-275, 2003.

[2] P. Ekman and E.L. Rosenberg, What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System. USA: Oxford University Press, 1997.

[3] C. Juanjuan, Z. Zheng, S. Han, and Z. Gang, "Facial expression recognition based on PCA reconstruction," in International Conference on Computer Science and Education, 2010, pp. 195-198.

[4] I. Cohen, N. Sebe, A. Garg, L.S. Chen, and T.S. Huang, "Facial expression recognition from video sequences: temporal and static modeling," Computer Vision and Image Understanding, vol. 91, no. 1, pp. 160-187, 2003.

[5] Agrawal, Shweta & Jain, Sanjiv. (2020). Medical Text and Image Processing: Applications, Issues and Challenges. Springer Nature 10.1007/978-3-030-40850-3_11.

[6] Shweta Agrawal, S. Jain, et al. "An orchestrator for networked control system and its application to collision avoidance in multiple mobile robots, International Journal of Engineering Systems Modelling and Simulation, 2021 Vol.12 No.2/3, pp.103 - 110

[7] Manoj Agrawal, Shweta Agrawal "Rice plant diseases detection classification using deep learning models: a systematic review" Journal of critical reviews JCR. 2020; vol. 7 issue: 11: 4376-4390

[8] M. Agrawal, S. Agrawal, "A Systematic Review on Artificial Intelligence/Deep Learning Applications and Challenges to battle against COVID-19 Pandemic" Disaster Advances, Vol. 14(8) August 2021, pp90-99.

[9] R. Tandon, S. Agrawal, "Sequential CNN for automatic breast cancer detection using histopathological images" Journal of critical reviews JCR 2020, Vol 7 issue 15: 6104-6117.

[10] D. Saravagi, S. Agrawal "Opportunities and challenges of ML model for prediction and diagnosis of spondylolithesis: A systematic review International Journal of Engineering Systems Modelling and Simulation, 2021 Vol.12 No.2/3, pp.127 - 138 (Scopus indexed)

[11] D. Saravagi, S. Agrawal "Indian Stock Market analysis and prediction using the LSTM model during COVID-19" International Journal of Engineering Systems Modelling and Simulation, 2021 Vol.12 No.2/3, pp.139 - 147 (Scopus indexed)

[12] M. Agrawal, S. Agrawal "Rice Plant Diseases Detection Using Convolutional Neural Networks" International Journal of Engineering Systems Modelling and Simulation, Accepted for publication (Scopus indexed)

[13] S. Jain, S. Agrawal "A Decision Tree C4.5 based Voltage Security Events Classifier for Electric Power Systems" International Journal of Engineering Systems Modelling and Simulation, Accepted for publication (Scopus indexed)

[14] Y. Wang, H. Ai, B. Wu, and C. Huang, "Real time facial expression recognition with adaboost," in Proceedings of the 17th International Conference on Pattern Recognition, 2004, pp. 926-929.

[15] D.T. Lin, "Facial expression classification using PCA and hierarchical radial basis function network," Journal of information science and engineering, vol. 22, no. 5, pp. 1033-1046, 2006.

[16] I.A. Essa and A.P. Pentland, "Coding, analysis, interpretation, and recognition of facial expressions," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 757-763, 1997.

[17] K. Anderson and P.W. McOwan, "A Real-Time Automated System for the Recognition of human facial expressions," IEEE Transactions on Systems, Man, and Cybernetics, vol. 36, no. 1, pp. 96-105, 2006.

[18] P. Kakumanu and N. Bourbakis, "A local-global graph approach for facial expression recognition," in IEEE International Conference on Tools with Artificial Intelligence, 2006, pp. 685-692.

[19] Shaik Asif Hussain, Ahlam Salim Abdallah al Balushi."A real time face emotion classification and recognition using deep learning model",journal of physics: conference series, 2020, publication.