A Project Report

On

Recognizing Human Face Expression Using CNN Algorithm in Deep Learning

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Technology Computer Science Engineering

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of

Mr. P. RAJA KUMAR

Assistant Professor

Submitted By

AJAY PRATAP SINGH (18SCSE1050042)

PRAKHAR JAIN (18SCSE1050023)

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA

INDIA

May, 2022

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project entitled **"Recognising Human Face Expression Using CNN Algorithm in Deep Learning"** in partial fulfillment of the requirements for the award of the the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of August 2021 to December 2021, under the supervision of "**Mr. P. RAJA KUMAR"**, Assistant Professor, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by us for the award of any other degree of this or any other places.

Ajay Pratap Singh
18SCSE1050042
Prakhar Jain
18SCSE10050023

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr. P. RAJA KUMAR
Assistant Professor

CERTIFICATE

The Final Project Viva-Voice examination of **Ajay Pratap Singh and Prakhar Jain** has been held on_____and his work is recommended for the award of Bachelor Of Technology in Computer Science and Engineering.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: May, 2022

Place: Greater Noida

ACKNOWLEDGEMENT

Apart from the efforts of our, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We would like to show my greatest appreciation to "**Mr. P. Raja Kumar"** and also our dean "**Dr. Munish Sabharwal"**. We can't say thank you enough for his tremendous support and help. We feel motivated and encouraged every time, We attend his meeting. Without his encouragement and guidance this project would not have materialized.

The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

Abstract

Human facial emotion recognition plays an important and vital role in the interpersonal relationship. Deep learning (DL) based emotion detection gives much better and exact results than traditional methods with image processing. It has been recognized for a many years and it is a vital topic in the fields of computer vision and machine learning. Deep Learning (DI) system is capable of emotion detection through facial expressions. By using Neural Network classifier training, five kinds of different emotional categories are obtained from images. Through this we proposed a Computer Vision (CV) based deep learning architecture for emotion detection from images. We collect the datasets from social media platforms and various websites. In this dataset, data will be presented in the form of image type with some facial expressions. The performance of the proposed method is evaluated using datasets for restaurants which are collected from different social media platforms and websites. Our aim is to understand the change in emotions of people through his face expressions.

Table of Contents

List of Table

List of Figure

Acronyms

B. Tech	Bachelor of Technology
SCSE	School of Computing Science and Engineering

CHAPTER-1

Introduction

1.1 Introduction:

The human face is an important part of an individual's body and it especially plays an important role in extraction of an individual's behavior and emotional state. As humans, we classify emotions all the time with/ without knowing it. In spite of their busyness if they see their facial expression then they may be try to do something different. For example, suppose if anyone see that his or her facial expression is happy then he or she will try to be happier. On the other hand, if anyone sees that his or her facial expression is sad then he or she will improve his or her mental condition. Facial expression plays an important role for detecting human emotion. It is a valuable indicator of a person. In a word an expression sends a message to a person about his or her internal feeling. Facial expression is the most important application of image processing. In the present age, a huge research works on the field of image processing. Facial image based emotion detection techniques provides a fast and useful result for emotion detection. The process of recognizing the expression of feelings through facial emotion was an interesting object. After 1960 this topic became more popular, when a list of universal emotion was established and different system was proposed. Because of the arrival of modern technology our expectation goes high and it has no limitation. As a result people try to improve this image based emotion detection in different ways. There are six basic universal emotions for human beings. These are happy, sad, angry, fear, disgust and surprise. From human's facial expression we can easily detect this emotion. In this project we will proposed a useful way to detect happy, sad and angry these three emotions from frontal facial emotion. Our aim, which we believe

we will develop a method of face emotion detection that is fast, robust, reasonably simple and accurate with a relatively simple and easy to understand algorithms and techniques. The examples provided in this thesis are real-time and taken from our own surroundings.

Image processing is a technique than can convert an image into digital form and perform different kinds of operation on it for getting better image and useful information. Image processing technique used two types of method. These are analog and digital image processing. Analog technique can be used for hard copies and digital technique used for manipulating digital image. The purpose of image processing is divided into five groups. These are: visualization, image sharpening, image retrieval, measurement of pattern, image recognition.

Visualization observe invisible object. Image sharpening that makes better image. Image retrieval finds interesting image. Measurement of pattern measures different objects of an image. Image recognition finds the difference of an image.

1.2 Problem Formulation:

Human facial expressions can be easily classified into 5 basic emotions: happy, sad, surprise, angry, and neutral. Our facial emotions are expressed through activation of specific sets of facial muscles. These sometimes subtle, yet complex, signals in an expression often contain an abundant amount of information about our state of mind.

Through facial emotion recognition, we are able to measure the effects that content and services have on the audience/users through an easy and low-cost procedure.

For example:

Retailers may use these metrics to evaluate customer interest.

Healthcare providers can provide better service by using additional information about patient's emotional state during treatment. Entertainment producers can monitor audience engagement in events to consistently create desired content.

Humans are well-trained in reading the emotions of others, in fact, at just 14 months old, babies can already tell the difference between happy and sad.

But can computers do a better job than us in accessing emotional states?

To answer this question, we are going to design deep learning neural network using computer vision that gives machines the ability to make inferences about our emotional states. In other words, we give them eyes to see what we can see.

Several Projects have already been done in this fields and our goal will not only be to develop a Automatic Facial Expression Recognition System but also improving the accuracy of this system compared to the other available systems.

Fig 1-Types of Emotions

1.3 Tools and Technology Used:

SOFTWARE REQUIREMENT:

As the project is developed in python, we have used Python 3.8 and Vscode and keras.

Keras

Keras is an Open Source Neural Network library written in Python that runs on top of Theano or Tensorflow. It is designed to be modular, fast and easy to use. It was developed by François Chollet, a Google engineer. Keras is high-level API wrapper for the low-level API, capable of running on top of TensorFlow, CNTK, or Theano. Keras High-Level API handles the way we make models, defining layers, or set up multiple input-output models. Deep learning is one of the major subfield of machine learning framework. Machine learning is the study of design of algorithms, inspired from the model of human brain. Deep learning is becoming more popular in data science fields like robotics, artificial intelligence(AI), audio & video recognition and image recognition. Artificial neural network is the core of deep learning methodologies. Deep learning is supported by various libraries such as Theano, TensorFlow, Caffe, Mxnet etc., Keras is one of the most powerful and easy to use python library, which is built on top of popular deep learning libraries like TensorFlow, Theano, etc., for creating deep learning models.Keras runs on top of open source machine libraries like TensorFlow, Theano or Cognitive Toolkit (CNTK). Theano is a python library used for fast numerical computation tasks. TensorFlow is the most famous symbolic math library used for creating neural networks and deep learning models. TensorFlow is very flexible and the primary benefit is distributed computing. CNTK is deep learning framework developed by

Microsoft. It uses libraries such as Python, C#, C++ or standalone machine learning toolkits. Theano and TensorFlow are very powerful libraries but difficult to understand for creating neural networks. Keras is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano. Keras is designed to quickly define deep learning models. Well, Keras is an optimal choice for deep learning applications. Keras leverages various optimization techniques to make high level neural network API easier and more performant. It supports the following features − Consistent, simple and extensible API. Minimal structure - easy to achieve the result without any frills. It supports multiple platforms and backends. It is user friendly framework which runs on both CPU and GPU. Highly scalability of computation.

Python

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python can be used on a server to create web applications. Python can be used alongside software to create workflows. Python can connect to database systems. It can also read and modify files. Python can be used to handle big data and perform complex mathematics. Python can be used for rapid prototyping, or for production-ready software development. Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).Python has a simple syntax similar to the English language. Python has syntax that allows developers to write programs with fewer lines than some other programming languages. Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that

prototyping can be very quick. Python can be treated in a procedural way, an object-oriented way or a functional way.

Vscode

Visual Studio Code is an source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Visual Studio Code has some very unique features. They are listed as below –

Support for multiple programming languages: Supports multiple programming languages. So earlier, programmers needed Web-Support: a different editor for different languages, but it has built-in multi-language support. This also means it easily detects if there's any fault or cross-language reference, it'll be able to detect it easily.

Intelli-Sense: It can detect if any snippet of code is left incomplete. Also, common variable syntaxes and variable declarations are made automatically. Ex: If a certain variable is being used in the program and the user has forgotten to declare, intelli-sense will declare it for the user.

Cross-Platform Support: Traditionally, editors used to support either Windows or Linux or Mac Systems. But Visual Studio Code is cross-platform. So it can work on all three platforms. Also, the code works on all three platforms; else, the open-source and proprietary software codes used to be different.

Extensions and Support: Usually supports all the programming languages but, if the user/programmer wants to use the programming language which is not supported then, he can download the extension and use it.

And performance-wise, the extension doesn't slow down the editor as it rums as a different process.

Repository: With the ever-increasing demand for the code, secure and timely storage is equally important. It is connected with Git or can be connected with any other repository for pulling or saving the instances.

Hierarchy Structure: The code files are located in files and folders. The required code files also have some files, which may be required for other complex projects. These files can be deleted as per convenience.

Improving Code: Some code snippets can be declared a bit differently, which might help the user in the code. This function prompts the user, wherever necessary, to change it to the suggested option.

Terminal Support: Many of the times, the user needs to start from the root of the directory to start with a particular action, in-built terminal or console provides user support to not to switch in-between two screens for the same.

Multi-Projects: Multiple projects containing multiple files/folders can be opened simultaneously. These projects/folders might or might not be related to each other.

Git Support: Resources can be pulled from Git Hub Repo online and vice-versa; saving can be done too. Resource pulling also means cloning the code which is made available on the internet. This code can later be changed and saved.

Commenting: A common feature, but some of the languages do not support it. Commenting on the code helps the user to recall or track according to the sequence he wants.

Deep Learning

Deep learning is based on the branch of machine learning, which is a subset of artificial intelligence. Since neural networks imitate the human brain and so deep learning will do. In deep learning, nothing is programmed explicitly. Basically, it is a machine learning class that makes use of numerous nonlinear processing units so as to perform feature extraction as well as transformation. The output from each preceding layer is taken as input by each one of the successive layers.

Deep learning models are capable enough to focus on the accurate features themselves by requiring a little guidance from the programmer and are very helpful in solving out the problem of dimensionality. Deep learning algorithms are used, especially when we have a huge no of inputs and outputs.

Since deep learning has been evolved by the machine learning , which itself is a subset of artificial intelligence and as the idea behind the artificial intelligence is to mimic the human behavior, so same is "the idea of deep learning to build such algorithm that can mimic the brain".

Deep learning is implemented with the help of Neural Networks, and the idea behind the motivation of Neural Network is the biological neurons, which is nothing but a brain cell.

“Deep learning is a collection of statistical techniques of machine learning for learning feature hierarchies that are actually based on artificial neural networks.”

So basically, deep learning is implemented by the help of deep networks, which are nothing but neural networks with multiple hidden layers.

- ❖ Features include:
    - editor with syntax highlighting and introspection for code completion
    - support for multiple Python consoles (including IPython)
    - the ability to explore and edit variables from a GUI
- ❖ Available plugins include:
    - Static Code Analysis with Pylint
    - Code Profiling
- ❖ Hardware Interfaces
    1. Processor : Intel CORE i5 processor with minimum 2.9 GHz speed.
    2. RAM : Minimum 4 GB.
    3. Hard Disk : Minimum 500 GB
- ❖ Software Interface
    1. Microsoft Word 2007
    2. VS code
    3. Operating System : Windows10

1.4 Motivation

In previous time, for psychologist, analyzing facial expression was an essential part. Nowadays image processing have motivated significantly on research work of automatic face mood detection. There are lots of depressed people lived in our society. Also lots of busy people those who do not know their present mental condition. So we try to develop such an application and by this application they will able to see their present mental condition.

CHAPTER-2

Literature Survey

As per various literature surveys it is found that for implementing this project four basic steps are required to be performed.

1. Preprocessing
2. Face registration
3. Facial feature extraction
4. Emotion classification

Description about all these processes is given below:

A) **Preprocessing :** is a common name for operations with images at the lowest Level of abstraction both input and output is intensity images. Most preprocessing steps that implemented are:-

   a. Reduce the noise.
   b. Convert The Image To Binary/Grayscale.
   c. Pixel Brightness Transformation.
   d. Geometric Transformation.

B) **Face Registration** : Face Registration is a computer technology being used in a variety of applications that identifies human faces in digital images. In this face registration step, faces are first located in the image using some set of landmark points called “face localization” or “face detection”. Thesedetected faces are then geometrically normalized to match some template image in a process called “face registration”.

C) **Facial Feature Extraction**: Facial Features extraction is an important step in face recognition and is defined as the process of locating specific regions, points, landmarks, or curves/contours in a given 2-D image or a 3D range image. In this feature extraction step, a numerical feature vector is generated from the resulting registered image. Common features that can be extracted area.

a. Lips

b. Eyes

c. Eyebrows

d. Nose tips

D) **Emotion Classification :** In the third step, of classification, the algorithm attempts to classify the given faces portraying one of the seven basic emotions. Paul Ekmanan American psychologist and professor who is a pioneer in the study of emotions and their relation to facial expressions. He has created an "atlas of emotions" with more than ten thousand facial expression.

2.1 Comparative Study:

Different approaches which are followed for Facial Expression Recognition:

Neural Network Approach : The neural network contained a hidden layer with neurons. The approach is based on the assumption that a neutral face image corresponding to each image is available to the system. Each neural network is trained independently with the use of on-line back propagation.

Principal of Component Analysis : Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variable called Principal Components.

Support Vector Machine : In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

Training & Testing Database: In machine learning, the study and construction of algorithms that can learn from and make predictions on data is a common task. Such algorithms work by making data driven predictions or decisions, through building a mathematical model from input data. The data used to build.

We will compare our research work with another research works and applications. We are actually study that research work and try to learn about them. We studied so many things from that research work and find many things such as algorithm, accuracy of that apps etc. all this research work are emotions analysis.

- "Infant facial expression and cries" in this paper author worked with infant

emotions. Author work with infants eyes and mouth

.author used clustering method, harmonic spectrum method. The accuracy of that Research is 75.2%.[1]

- "Emotion recognition by dynamic HOG features" in this paper author elaborated an emotion recognition framework using dynamic dense grid-based HoG features. Proposed method performs better than static Uniform LBP implementation, used in the "baseline method" offered by the challenge organizers. The accuracy of emotion detection is 70%.[2]

- "Emotion based music player for android" in this paper author worked with emotion. Author work with viola and Jones algorithm, support vector machines (SVM) method.[3]

- Emotion recognition system for mobile application" in this paper author work with facial emotions. Author use viola and Jones algorithm .in this research author found 92.7% accuracy.[3]

- "Automatic Recognition of Facial Displays of Unfelt Emotions" in this paper author works with emotions. Author was worked with simulation theory and also works with felt and unfelt emotions. After testing 51% accuracy was gain.[5]

- "An Emotion Recognition Challenge" in this paper authorused baseline method, principal component analysis (PCA) algorithm. The accuracy of thisresearch work is 62.3%.[4]

- "Mood Prediction from Facial Video with Music "Therapy" on a Smartphone" This paper presents a prototype desktop version and a Smartphone app which analyses the mood of a video and predict the user's mood. The app can play songs and change the song according to the mood analyzed. Accuracy of this work is

60%.[8]

Image Processing is a useful method for performing different operations on an image to get a better image or to get some useful information from it. Normally image processing method consider an images as a two dimensional signals. Because of this usefulness of image processing, in our research we are dealing with this method. Mainly the project aim is to detect human's facial expression by applying image processing techniques and send them a massage about their internal feelings based on their facial expression. Those people who remain submerged in despair, this application are more beneficial for them. This application can get rid of their stress by playing music or jokes automatically. We hope that this application will bring a significant change of human life.

2.2 EXISTING SYSTEM:

There have been constant improvements in image processing and emotion recognition techniques over the past years but yet some constraints still exist in the present systems which refrain from getting efficient images recognitions. Some of the constraints are:

- Low accuracy- Face recognition has low accuracy compared to the proven performance of finger print and iris recognition.

- Variety of images of single faces- There are many attributes

leading to the variability of images of a single face that add to the complexity of the recognition problem if they cannot be avoided by careful design of the capture situation.

- Personal changes- Change in facial expressions, aging and other personal factors also add to the difficulty in recognizing faces.

- Camera variations- Different cameras carry different lenses which again has a great impact on the picture being captured as different lens have different power. So any change in the camera or the lenses adds to the difficulty.

- FALSE ACCEPT- Impersonating somebody else is not a difficult task anymore. It can be easily achieved in today's time. This imitation if perfectly done by the attacker can easily con the system. This is called FALSE ACCEPT, Similarly many emotions and expressions can be wrongly classified sometimes.

- FALSE REJECT- A little expression or accessory can change the whole look of a person. This leads to unnecessary dismissal of the same face. This is called FALSE REJECT.

2.3 PROPOSED SYSTEM:-

There have been several advances in the past few years in terms of face detection, feature extraction mechanisms and the techniques used for expression classification, but development of an automated system that accomplishes this task is difficult. In this paper, we present an approach based on Convolutional Neural Networks (CNN) for facial expression recognition. The input into our system is an image; then, we use CNN to predict the facial expression label which should be one these labels: anger, happiness, fear, sadness, disgust and neutral.

My goal for this project is:

- Handle partial occlusions better.
- Make it more robust (lighting conditions etc.)
- More person independent
- Apply emotion recognition in applications. For example games,entertainment.

2.4 ARCHITECTURE DIAGRAM FOR PROPOSED METHOD:-

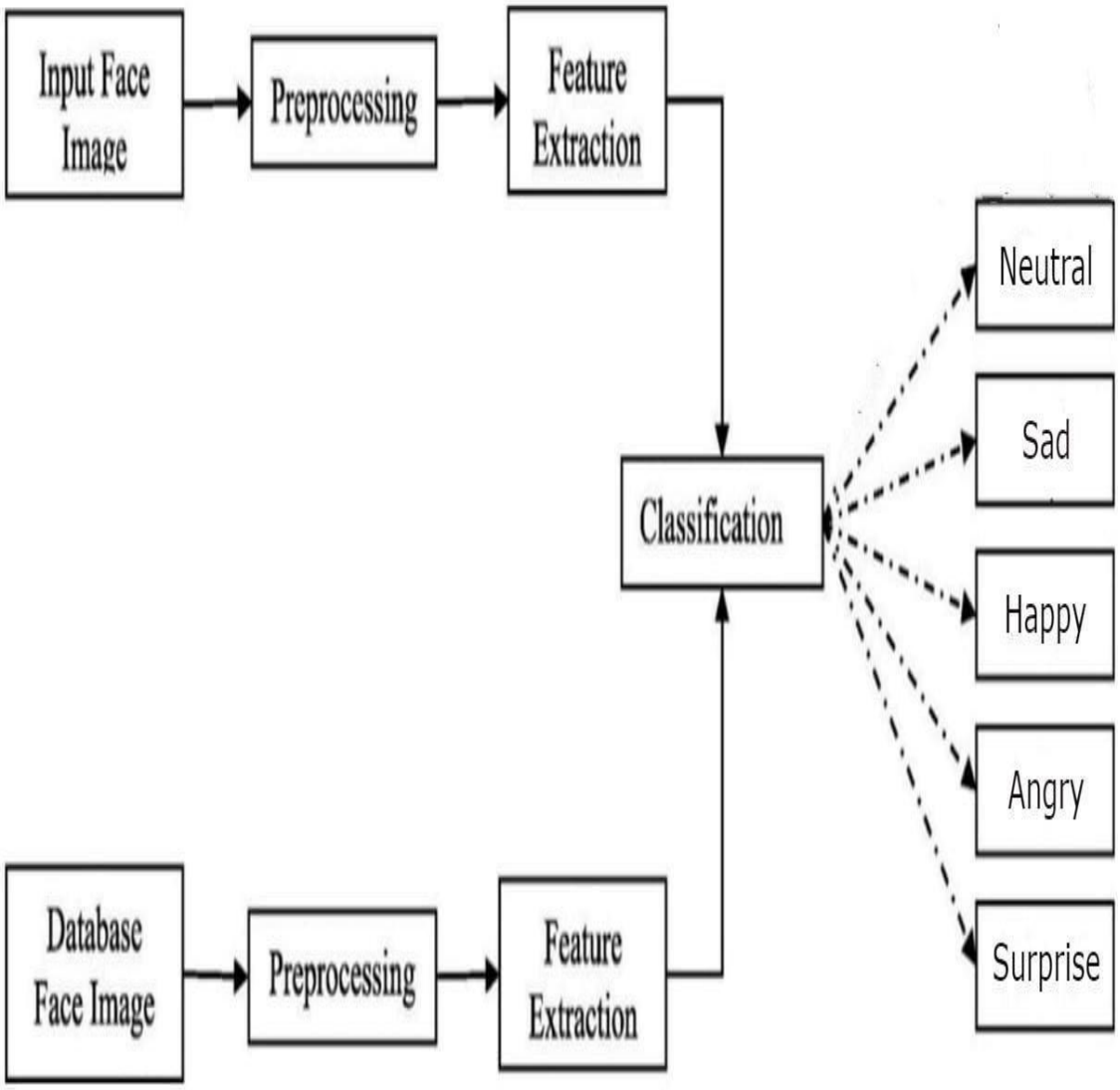


Fig-2 Face Recognition Architecture Diagram

3 : PREREQUISITES OF THE PROJECT:

3.1 Introduction:-

Image processing is a technique than can convert an image into digital form and perform different kinds of operation on it for getting better image and useful information. Image processing technique used two types of method. These are analog and digital image processing. Analog technique can be used for hard copies and digital technique used for manipulating digital image. The purpose of image processing is divided into five groups. These are: visualization, image sharpening, image retrieval, measurement of pattern, image recognition. Visualization observe invisible object. Image sharpening that makes better image. Image retrieval finds interesting image. Measurement of pattern measures different objects of an image. Image recognition finds the difference of an image.

3.1.1 Machine Learning Algorithms:-

One of the most important applications of artificial intelligence is machine learning. It provides the application that can automatically learn and improve from experience without being apparently programmed. The learning process starts with observations or data. Such as, we can assume a good decision based on direct experience or instruction. The basic aim is to allow the device without human interruption.

a) Supervised Machine Learning Algorithms:

Supervised learning algorithms able to do different analysis with new data based on what it learned from the past and can also predict future event. The supervised learning algorithms create deduced function for predicting the starting analysis of known training data and output values. After some effective training the system makes a target for any new inputs. The system is able to compare its output with correct output and also find error for modification.

b) Unsupervised Machine Learning Algorithms:

Unsupervised machine learning algorithms are used for training unclassified and those data which are not leveled. Unsupervised learning is able to describe a secret shape from unleveled data. This system can't provide proper output but it is able to take important decision from data set for describing secret shape from unleveled data.

c) Semi-supervised Machine Learning Algorithms:

Semi-supervised machine algorithms lies between supervised and unsupervised learning. For training they use both leveled and unleveled data but in this training data there are a small amount of leveled data and a huge amount of unleveled data. By using this method the systems are able to develop learning exactitude. Normally semi-supervised learning algorithms are used when leveled data need proficient and relevant resource for training.

3.1.2 DEEP LEARNING:-

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It's achieving results that were not possible before.

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

Deep learning achieves recognition accuracy at higher levels than ever before. This helps consumer electronics meet user expectations, and it is crucial for safety- critical applications like driverless cars. Recent advances in deep learning have improved to the point where deep learning outperforms humans in some tasks like classifying objects in images.

3.2 FUNCTIONAL REOUIREMENTS:-

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases.

Here, the system has to perform the following tasks:

- Take the real time input of the person from the web cam.
- Identify the face and extract the facial features.
- Based on the trained data, classify the emotion/gender and if any Objects seen.
- The recognized emotion is given as output in the form of a speech

3.3 NON-FUNCTIONAL REQUIREMENTS:-

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "non-behavioral requirements". Some of the quality attributes are as follows:

ACCESSIBILITY:-

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible.

The system will be accessible to a lot of people as it will be incorporated with virtual assistants and home service robots.

MAINTAINABILITY:-

In software engineering, maintainability is the ease with which a software product can be modified in order to:

- Correct defects
- Meet new requirements

Since the project will be implemented using python libraries it is easier to add or modify the code since it won't have a larger code.

SCALABILITY:-

System is capable of handling increase total throughput under an increased load when resources (typically hardware) are added. System can work normally under situations such as low bandwidth and large number of users.

PORTABILITY:-

Portability is one of the key concepts of high-level programming. Portability is the software code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another. Project can be executed under different operation conditions provided it meet its minimum configurations. Only system files and dependant assemblies would have to be configured in such case.

3.4 HARDWARE REQUIREMENTS:-

Processor : Any Processor above 500 MHz

RAM : 512Mb

Hard Disk : 10 GB

Input device : Standard Keyboard & Mouse, WebcamOutputdevice

: VGA and High Resolution Monitor

3.5 SOFTWARE REQUIREMENTS:-

Operating system : Windows XP or above

Language : Python

IDE : VSCODE

4:- Methodology

4.1 CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Network is one of the technique to do image classification and image recognition in neural networks. It is designed to process the data by multiple layers of arrays. This type of neural network is used in applications like image recognition or face recognition.

The primary difference between CNN and other neural network is that CNN takes input as a two-dimensional array. And it operates directly on the images rather than focusing on feature extraction which other neural networks do.

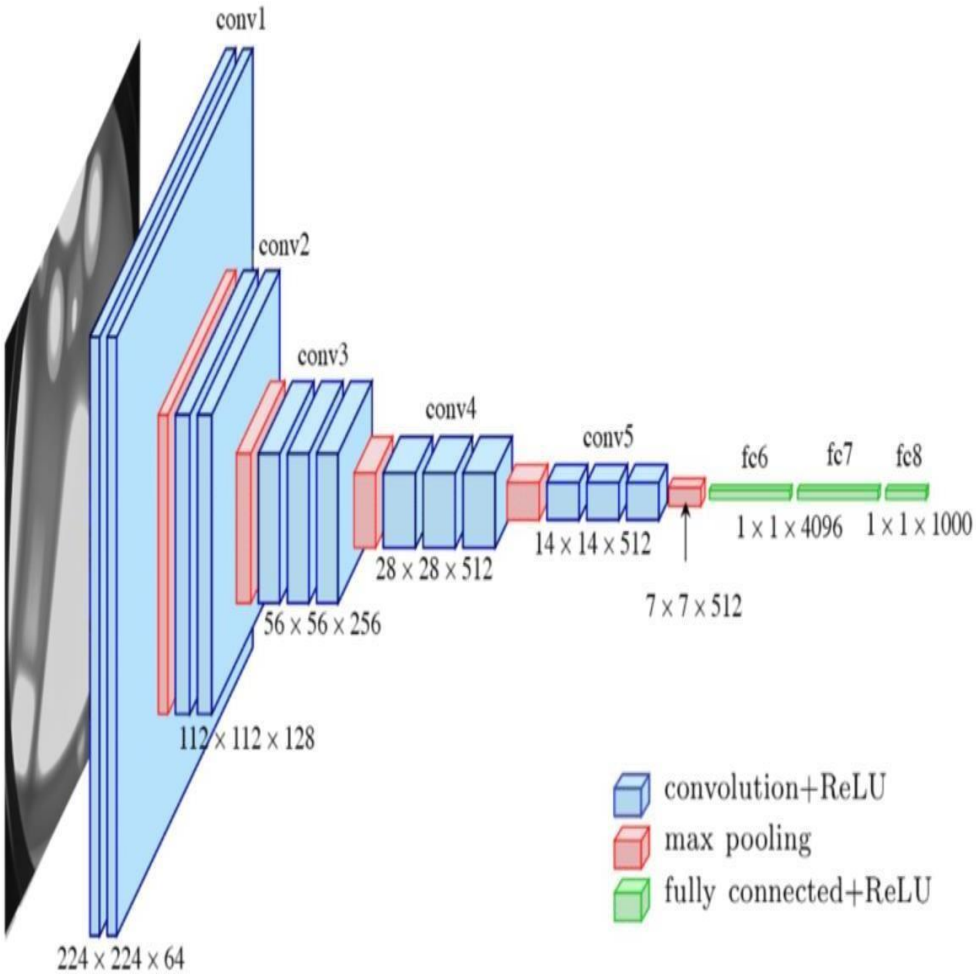


Fig 3:- Architecture Of CNN Model

Convolutional Neural Networks have the following 4 layers:

- Convolutional
- ReLU Layer
- Pooling
- Fully Connected

Convolutional layer

Convolution layer is the first layer to derive features from the input image. The convolutional layer conserves the relationship between pixels by learning image features using a small square of input data. It is the mathematical operation which takes two inputs such as image matrix and kernel or any filter.

The dimension of image matrix is h×w×d.

The dimension of any filter is fh×fw×d.

The dimension of output is (h-fh+1)×(w-fw+1)×1.

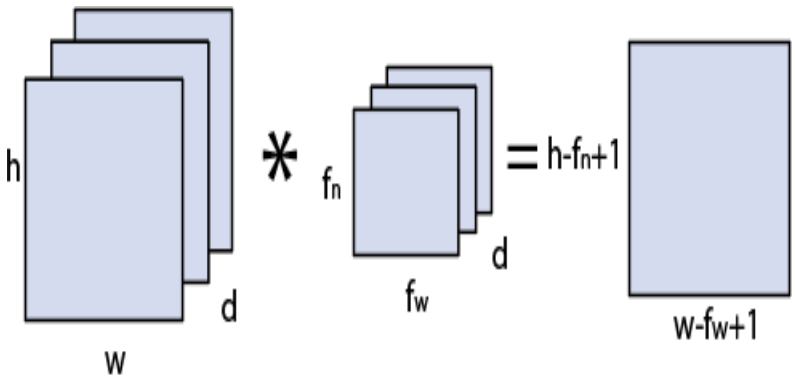


Fig 4 : Filter Matrix

ReLU Layer

Rectified Linear unit(ReLU) transform functions only activates a node if the input is above a certain quantity. While the data is below zero, the output is zero, but when the input rises above a certain threshold. It has a linear relationship with the dependent variable.

In this layer, we remove every negative value from the filtered images and replaces them with zeros.

It is happening to avoid the values from adding up to zero.

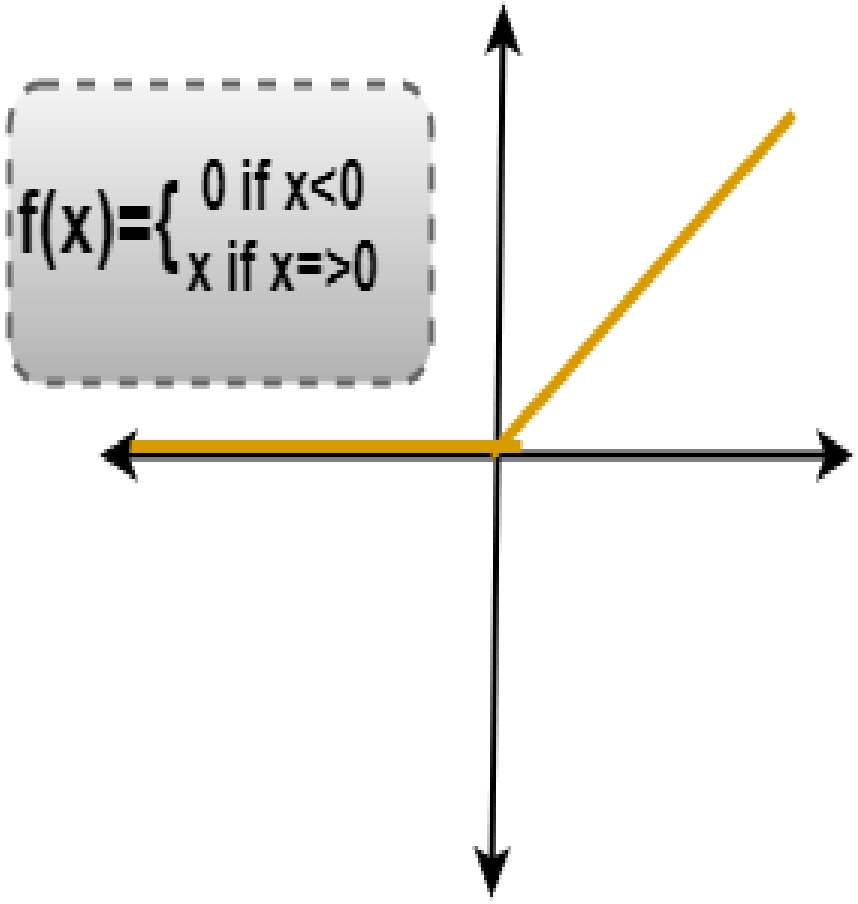


Fig 5 : ReLU function

Pooling Layer

Pooling layer plays a vital role in pre-processing of any image. Pooling layer reduces the number of the parameter when the image is too large. Pooling is "downscaling" of the image achieved from previous layers. It can be compared to shrink an image to reduce the image's density. Spatial pooling is also called downsampling and subsampling, which reduce the dimensionality of each map but remains essential information. These are the following types of spatial pooling.

We do this by implementing the following 4 steps:

Pick a window size (usually 2 or 3)

Pick a stride (usually 2)

Walk your window across your filtered images

From each window, take the maximum value

Max Pooling

Max pooling is a sample-based discretization process. The main objective of max-pooling is to downscale an input representation, reducing its dimension and allowing for the assumption to be made about feature contained in the sub-region binned.

Max pooling is complete by applying a max filter in non-overlapping sub-regions of initial representation.

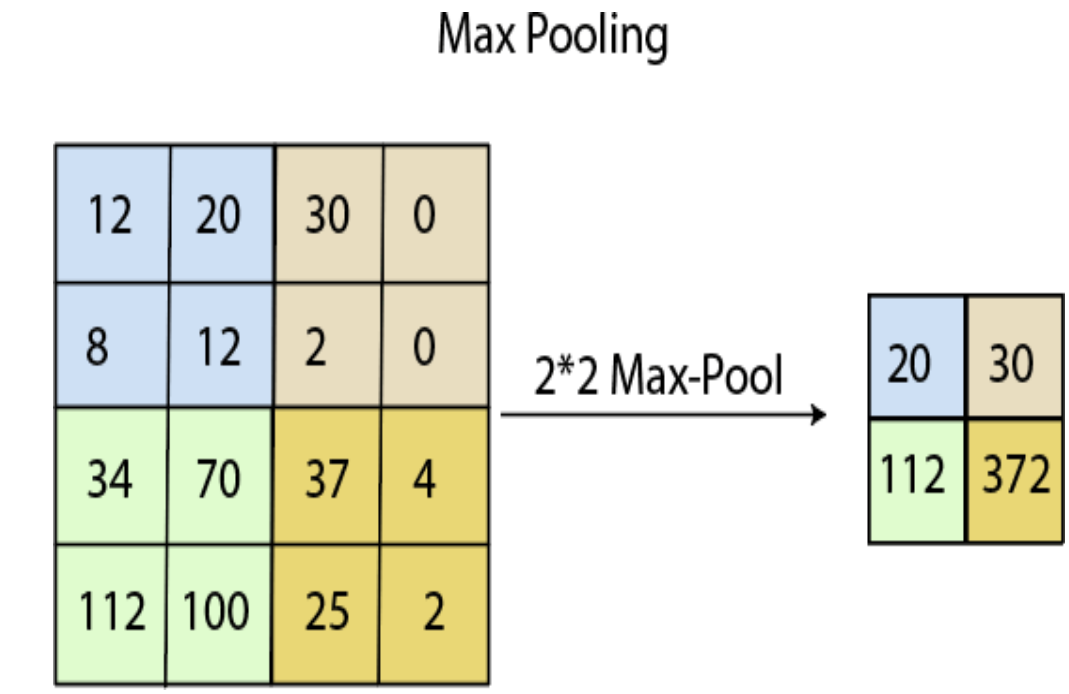


Fig 6 : Max Pooling(a)

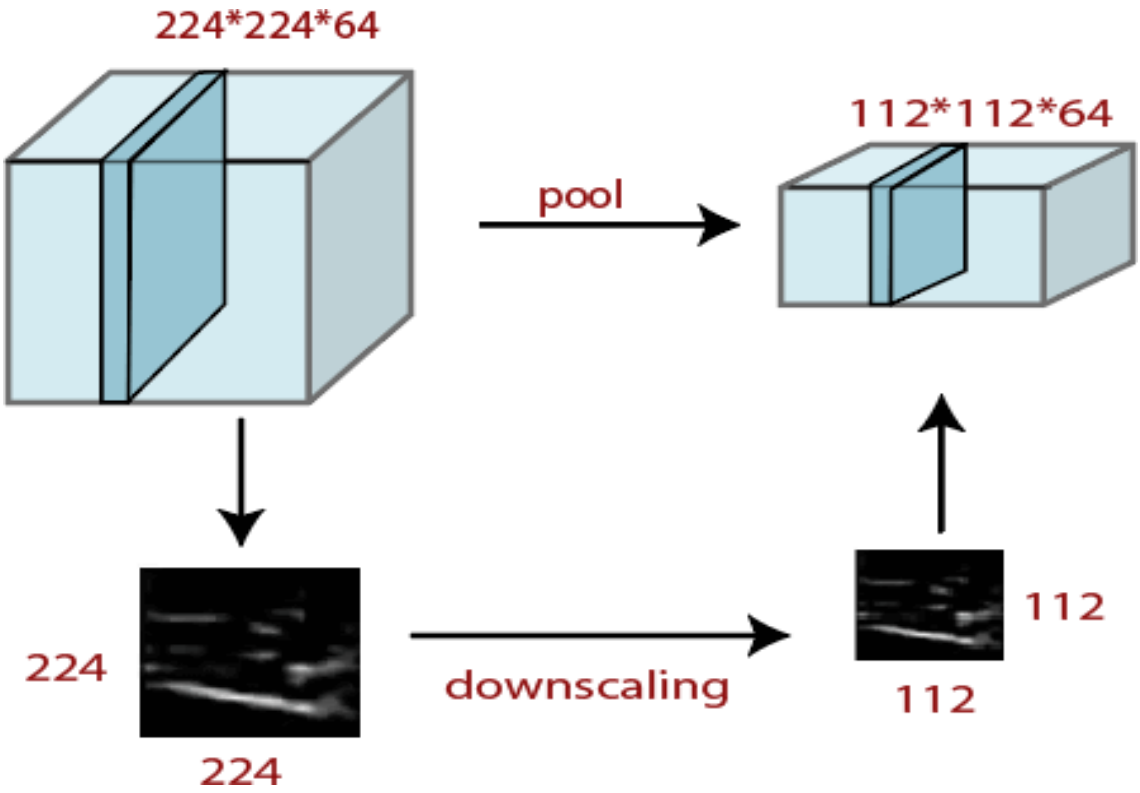


Fig 7 : Max Pooling(b)

Average Pooling

Down-scaling will perform by average pooling by dividing the input into rectangular pooling regions and computing the average values of each area.

Syntax

layer = averagePooling2dLayer(pool Size)

layer = averagePooling2dLayer(poolSize, Name, Value)

Sum Pooling

The sub-region for sum pooling and mean pooling are set the same as for max-pooling but instead of using the max function we use sum or mean.

In this layer we shrink the image stack into a smaller size steps;

Pick a window size (usually 2 or 3)

Pick a stride (usually 2)

Walk our window across our filtered images.

From each window, take the maximum value.

Performing pooling with a window size two and stride 2

Fully Connected (Dense) Layer

The fully connected layer (dense layer) is a layer where the input from other layers will be depressed into the vector. It will transform the output into any desired number of classes into the network.

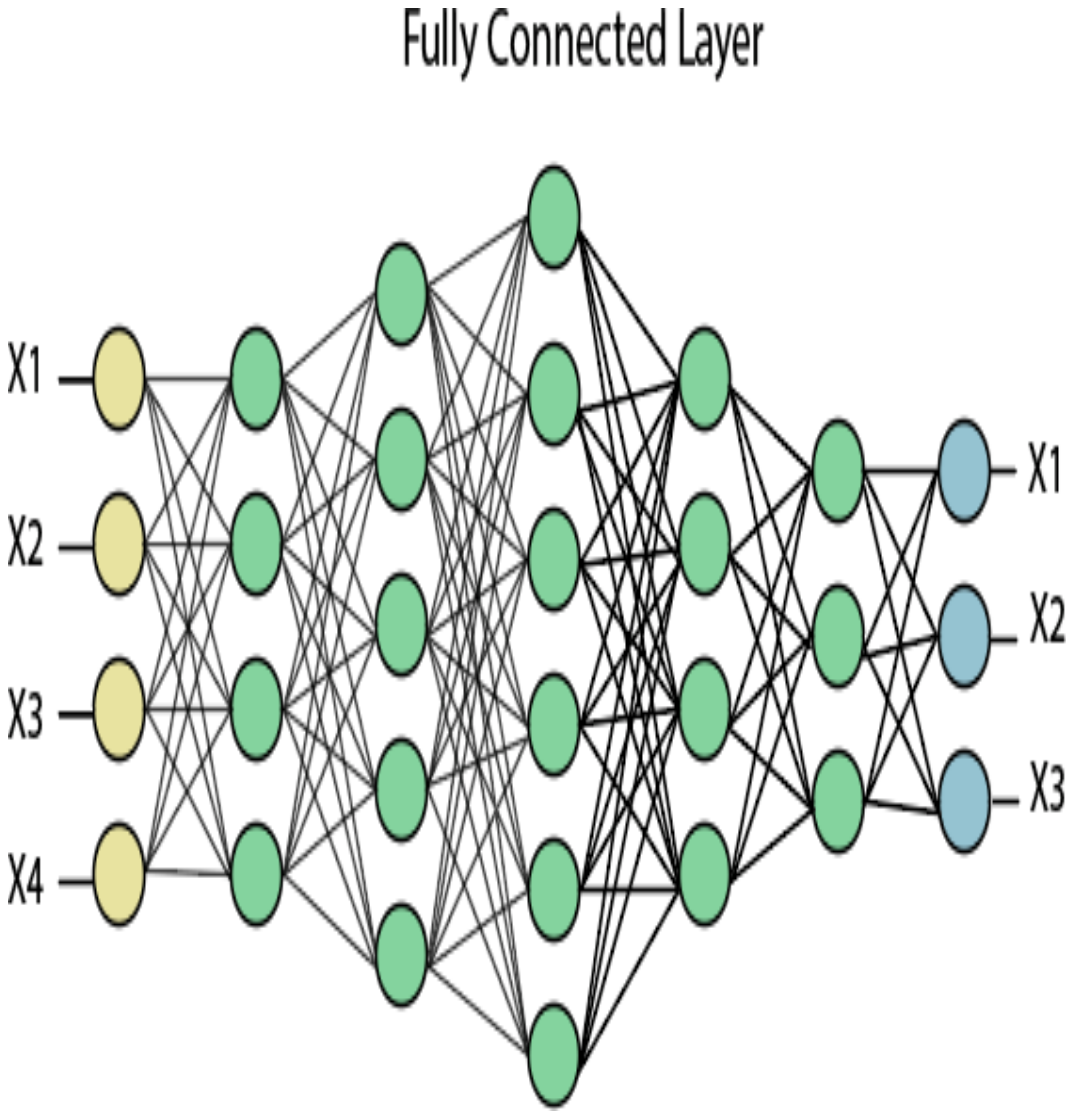


Fig 8 : Fully Connected Layer

In the above diagram, the map matrix is converted into the vector such as x1, x2, x3... xn with the help of a fully connected layer. We will combine features to create any model and apply activation function like as softmax or sigmoid to classify the outputs as a car, dog, truck, etc.

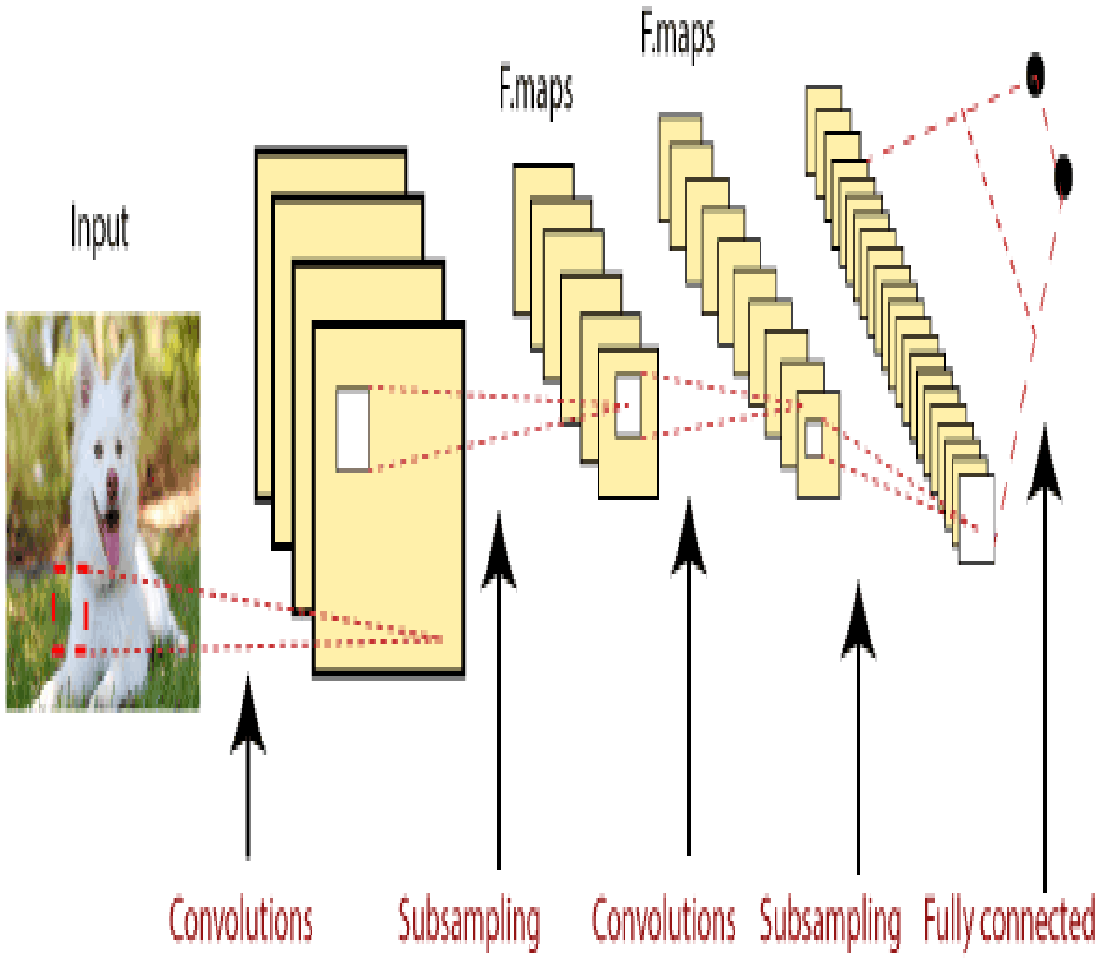


Fig 9 : Image Classification

This is the final where the actual classification happens.

Structure of Our CNN Model

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 48, 48, 32)	320
activation (Activation)	(None, 48, 48, 32)	0
batch_normalization (BatchNo	(None, 48, 48, 32)	128
conv2d_1 (Conv2D)	(None, 48, 48, 32)	9248
activation_1 (Activation)	(None, 48, 48, 32)	0
batch_normalization_1 (Batch	(None, 48, 48, 32)	128
max_pooling2d (MaxPooling2D)	(None, 24, 24, 32)	0
dropout (Dropout)	(None, 24, 24, 32)	0
conv2d_2 (Conv2D)	(None, 24, 24, 64)	18496
activation_2 (Activation)	(None, 24, 24, 64)	0
batch_normalization_2 (Batch	(None, 24, 24, 64)	256
conv2d_3 (Conv2D)	(None, 24, 24, 64)	36928
activation_3 (Activation)	(None, 24, 24, 64)	0
batch_normalization_3 (Batch	(None, 24, 24, 64)	256
max_pooling2d_1 (MaxPooling2	(None, 12, 12, 64)	0
dropout_1 (Dropout)	(None, 12, 12, 64)	0
conv2d_4 (Conv2D)	(None, 12, 12, 128)	73856
activation_4 (Activation)	(None, 12, 12, 128)	0
batch_normalization_4 (Batch	(None, 12, 12, 128)	512
conv2d_5 (Conv2D)	(None, 12, 12, 128)	147584
activation_5 (Activation)	(None, 12, 12, 128)	0
batch_normalization_5 (Batch	(None, 12, 12, 128)	512
max_pooling2d_2 (MaxPooling2	(None, 6, 6, 128)	0
dropout_2 (Dropout)	(None, 6, 6, 128)	0
conv2d_6 (Conv2D)	(None, 6, 6, 256)	295168
activation_6 (Activation)	(None, 6, 6, 256)	0
batch_normalization_6 (Batch	(None, 6, 6, 256)	1024
conv2d_7 (Conv2D)	(None, 6, 6, 256)	590080
activation_7 (Activation)	(None, 6, 6, 256)	0
batch_normalization_7 (Batch	(None, 6, 6, 256)	1024
max_pooling2d_3 (MaxPooling2	(None, 3, 3, 256)	0
dropout_3 (Dropout)	(None, 3, 3, 256)	0
flatten (Flatten)	(None, 2304)	0
dense (Dense)	(None, 64)	147520
activation_8 (Activation)	(None, 64)	0
batch_normalization_8 (Batch	(None, 64)	256
dropout_4 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 64)	4160
activation_9 (Activation)	(None, 64)	0
batch_normalization_9 (Batch	(None, 64)	256
dropout_5 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 5)	325
activation_10 (Activation)	(None, 5)	0

Total params: 1,328,037
Trainable params: 1,325,861

Table 1: Structure of CNN model

5 : Working of Project

The system has achieved an overall accuracy of **69%** after training it for 25 epochs on a Windows system. The model can detect multiple faces in real time. Firstly, it identifies the emotion in each of the detected faces and then labels the emotion accordingly near the bounding box that appears around each detected face. The decision to build this on Python gives us the versatility to extend the project beyond mere facial emotion detection. We can build various applications with this project at the heart of each one of them.

→About Data set :

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

The task is to categorize each face based on the emotion shown in the facial expression into one of five categories (1=Neutral, 2=Happy, 3=Sad, 4=Angry, 5=Surprise). The training set consists of 24,176 examples and the validated test set consists of 3,006 examples.

6:- Implementation

6.1 Code

Classification_little_vgg.py

```
from__future__import print_function
import keras
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import
Dense,Dropout,Activation,Flatten,BatchNormalization
from keras.layers import Conv2D,MaxPooling2D
import os

num_classes = 5
img_rows,img_cols = 48,48
batch_size = 32

train_data_dir = r'C:\Users\Acer\Desktop\Live Project\train'
validation_data_dir = r'C:\Users\Acer\Desktop\Live Project\validation'

train_datagen = ImageDataGenerator(
                                    rescale=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,
                                    fill_mode='nearest')

validation_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
                                    train_data_dir,
                                    color_mode='grayscale',
                                    target_size=(img_rows,img_cols),
                                    batch_size=batch_size,
                                    class_mode='categorical',
                                    shuffle=True)

validation_generator = validation_datagen.flow_from_directory(
                                            validation_data_dir,
                                            color_mode='grayscale',

    target_size=(img_rows,img_cols),
                                            batch_size=batch_size,
                                            class_mode='categorical',
                                            shuffle=True)
```

```
model = Sequential()

# Block-1

model.add(Conv2D(32,(3,3),padding='same',kernel_initializer='he_normal',
input_shape=(img_rows,img_cols,1)))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Conv2D(32,(3,3),padding='same',kernel_initializer='he_normal',
input_shape=(img_rows,img_cols,1)))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

# Block-2

model.add(Conv2D(64,(3,3),padding='same',kernel_initializer='he_normal'
))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Conv2D(64,(3,3),padding='same',kernel_initializer='he_normal'
```

```
))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

# Block-3

model.add(Conv2D(128,(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Conv2D(128,(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

# Block-4

model.add(Conv2D(256,(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
```

```
model.add(BatchNormalization())
model.add(Conv2D(256,(3,3),padding='same',kernel_initializer='he_norma
l'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

# Block-5

model.add(Flatten())
model.add(Dense(64,kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))

# Block-6

model.add(Dense(64,kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))

# Block-7
```

```
model.add(Dense(num_classes,kernel_initializer='he_normal'))
model.add(Activation('softmax'))

print(model.summary())
from tensorflow.keras.optimizers import RMSprop,SGD,Adam
# from keras.optimizers import SGD,Adam
from keras.callbacks import ModelCheckpoint, EarlyStopping,
ReduceLROnPlateau

checkpoint = ModelCheckpoint(r'C:\Users\Acer\Desktop\Live
Project\Emotion_little_vgg.h5',
                             monitor='val_loss',
                             mode='min',
                             save_best_only=True,
                             verbose=1)

earlystop = EarlyStopping(monitor='val_loss',
                          min_delta=0,
                          patience=3,
                          verbose=1,
                          restore_best_weights=True
                          )
```

```
reduce_lr = ReduceLROnPlateau(monitor='val_loss',
                              factor=0.2,
                              patience=3,
                              verbose=1,
                              min_delta=0.0001)

callbacks = [earlystop,checkpoint,reduce_lr]

model.compile(loss='categorical_crossentropy',
              optimizer = Adam(lr=0.001),
              metrics=['accuracy'])

nb_train_samples = 24176
nb_validation_samples = 3006
epochs=25

history=model.fit_generator(
                train_generator,
                steps_per_epoch=nb_train_samples//batch_size,
                epochs=epochs,
                callbacks=callbacks,
                validation_data=validation_generator,
                validation_steps=nb_validation_samples//batch_size)
```

Facial_expressions_recog.py

```
from keras.models import load_model
from time import sleep

from keras.preprocessing.image import img_to_array
from keras.preprocessing import image
import cv2
import numpy as np

face_classifier = cv2.CascadeClassifier(r'C:\Users\Acer\Desktop\Live
Project\haarcascade_frontalface_default.xml')

classifier =load_model(r'C:\Users\Acer\Desktop\Live
Project\Emotion_little_vgg.h5')

class_labels = ['Angry','Happy','Neutral','Sad','Surprise']
cap = cv2.VideoCapture(0)

while True:
    # Grab a single frame of video
    ret, frame = cap.read()
    labels = []
    gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces = face_classifier.detectMultiScale(gray,1.3,5)
```

```
    for (x,y,w,h) in faces:
        cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
        roi_gray = gray[y:y+h,x:x+w]
        roi_gray =
cv2.resize(roi_gray,(48,48),interpolation=cv2.INTER_AREA)
    # rect,face,image = face_detector(frame)



        if np.sum([roi_gray])!=0:
            roi = roi_gray.astype('float')/255.0
            roi = img_to_array(roi)
            roi = np.expand_dims(roi,axis=0)


        # make a prediction on the ROI, then lookup the class


            preds = classifier.predict(roi)[0]
            label=class_labels[preds.argmax()]
            label_position = (x,y)

cv2.putText(frame,label,label_position,cv2.FONT_HERSHEY_SIMPL


EX,2,(0,255,0),3)
```

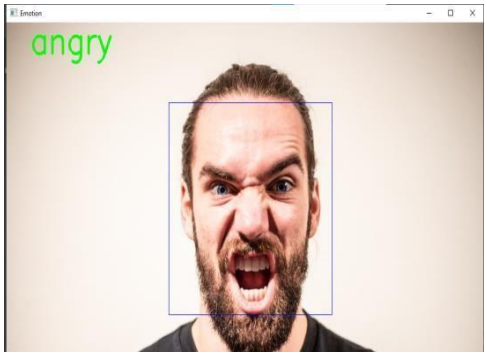
```
        else:
            cv2.putText(frame,'No Face
Found',(20,60),cv2.FONT_HERSHEY_SIMPLEX,2,(0,255,0),3)
    cv2.imshow('Emotion  Detector',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break


cap.release()

cv2.destroyAllWindows()
```

6.2 OUTPUT

It provides accurate result for happy facial mood expression. And most of the time it provides all most accurate result for angry facial mood expression. But the application faces some problems for detecting calm and sad facial mood expression.

At first, user needs to take an image as input. For improving lost contrast, use histogram equalization by remapping the brightness value of an image. Then detect face boundary, cropping eye and cropping lip region by PCA and MPCA. Then it sends the image to machine learning kit. Machine learning kit is recently developed by google which has trained data. It provides powerful feature and bear new information. That's why machine learning becomes most popular nowadays. Machine learning SDK can recognize text, detect faces, recognize landmarks, scan bar codes and leveling images. In this project we use ML kit for detecting face mood. It can detect facial expression percentage. But applying some conditions, using ML kit we develop five facial expressions (Happy, Sad, Neutral, Surprise and Angry) and on the basis of expression it provide the facial expression.

After selecting an image as input PCA and MPCA analyze this image and send it to the machine learning kit for further work. Then using ML kit and applying some condition we get almost 70% accurate results about human's facial expression.



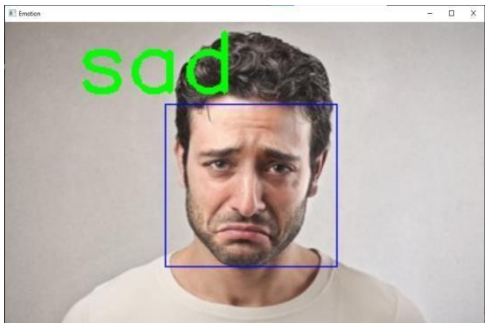


Fig 11: Sample Outputs

7:- Conclusion and future scope

We have uses the approach of convolution neural network with keras for face emotion recognition. Approx we have trained 24176 images samples and validated 3006 images sample. As we have uses the dataset image of 48*48 sizes (grayscale) and using the face expression recognition dataset we have achieved the accuracy of 69 % in our trained convolutional neural network model. Our trained model is able to detect the different emotions like neutral, happy, sad, angry and sad. We can increase the accuracy of model or we can say that we can make a better trained model through increasing the number of epochs up to a limit but we can't make the accuracy exactly 100%.

And in future train the same convolutional neural network model with different dataset i.e. the effective dataset as well as with different number of epochs and see what accuracy will be achieved by changing these factors.

8 :- References

1- Javatpoint Deep Learning, CNN.

2- Apeksha Khopkar and Ashish Adholiya Saxena, Facial expression recognition using Keras, Biotech Research Communication 14(5), 2021.

3- Enkhzul Enkhtaivan, Tosin A. Adesuyi, Sunmyeng Kim, "Facial Emotion Recognition using Convolutional Neural Network Based on Repetitive Learning Blocks Approach" Department of Software Engineering, Kumoh National Institute of Technology, Gumi, South Korea.

4- I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner,W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio. Challenges in representation learning: A report on three machine learning contests. Neural Networks, 64:59–63, 2015. Special Issue on "Deep Learning of Representations".

5- Vijaya Prasad K., Kishore P.V.V., Srinivasa Rao O. (2019), 'Skeleton based view invariant human action recognition using convolutional neural networks', International Journal of Recent Technology and Engineering, 8(2), PP.4860-4867.

6- Aya Hassouneh a, A.M. Mutawa a , M. Murugappan, “Development of a Real-Time Emotion Recognition System Using Facial Expressions and EEG based on machine learning and deep neural network methods” Informatics in Medicine Unlocked 20 (2020) 100372.

7- Vepuri, K., Attar, N. (2021), 'Improving the Performance of Deep Learning in Facial Emotion Recognition with Image Sharpening', World Academy of Science, Engineering and Technology, Open Science Index 172, InternationalJournal of Computer, and Information Engineering, 15(4), 234 – 237.

8- Kruti Goyal, KartikeyAgarwal, Rishi Kumar CSE, Amity, University Noida, India. Face Detection and Tracking Using OpenCV. International Conference on Electronics, Communication, and Aerospace Technology ICECA 2017.

9- Prasad M.V.D., Lakshmamma B.J., Chandana A.H., Komali K., Manoja M.V.N., Kumar P.R., Prasad C.R., Inthiyaz S., Kiran P.S. (2018) , ‘An efficient classification of flower images with convolutional neural networks’ ,International Journal of Engineering and Technology(UAE), 7 (1.1),PP. 384- 391.

10- A. A. Gunawan et al. Face expression detection on kinect

usingactive appearance model and fuzzy logic. Procedia Computer Science,59:268–274, 2015.

11- Meriem Sari , Abdelouahab Moussaoui, and Abdenour Hadid "Automated Facial Expression Recognition Using Deep Learning Techniques: An Overview" International Journal of Informatics and Applied Mathematics e- ISSN:2667-6990 Vol. 3, No. 1, 39-53.