ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

FACE EMOTION DETECTION USING DEEP LEARNING

K. Latha, S. Mounika, Ch. Rajesh, T. Raghunath, K. Dinesh and T. V. S. Sriram
Nadimpalli Satyanarayana Raju Institute of Technology, Visakhapatnam– 531173, Andhra Pradesh, India
lathakappa2001@gmail.com, saripallimounika3@gmail.com, rajeshchukka12345@gmail.com, ravanraghunath@gmail.com, dineshshankar2403@gmail.com and tvssriram.cse@nsrit.edu.in

Abstract
In our lives emotions are the fundamentals for human beings and play an important role in human cognition. Expressing and recognizing emotions of human are very much important in communication system. Emotions can be expressed through gestures, speech, facial expressions, body language etc. Facial movement is the major role in expressing emotions. Expressing emotions varies from one person to another. Human emotions are classified as: surprise, fear, anger, happy, sad, disgust andneutral.
Emotions effect human both positively and negatively and at time applies on healthof an individual. People while in emotions need to come back to normalcy as quick and as easily as possible. In case of negative emotions (that effect negatively on body) need immediate treatment. Traditionally there are several ways to treat they include listening to music, reading a book, listening to humor, watching a movie , talking to a friend etc., Though these are good ways to overcome emotions, time and place (location) in today's world may have limitations. In today's digital world and days of smart phones we can overcome place and time factors. This necessitated us to think and develop a "Recommendation system" on the state of art technologies and concepts like "DEEP LEARNING". We come up with a solution in the form of "FACE EMOTION DETECTION USING DEEP LEARNING (FED)".
FED aims to personalize emotions driven recommendation of entertainment. To start with a case we have chosen Music as it has a great influence on humans and is widely used for relaxing, mood regulation, destruction from stress, to maintain mental and physical work.

FED is designed to the personalized music recommendation system driven by listener feelings, emotions and activity contexts. FED automates the capture and treatment of emotions by using classification algorithm and "DEEP LEARNING" techniques that categorizing the facial expressions. Though FED is presently focusing on Music, this can be further extended to several other entertainment types and solutions.

Keywords: Face emotion detection, Deep learning, Automates the capture treatment and Music.

Cite this article as: K.Latha, S.Mounika, Ch.Rajesh, T.Raghunath, K.Dinesh & T.V.S.Sriram, "FACE EMOTION DETECTION USING DEEP LEARNING", International Journal of Research in Advanced Computer Science Engineering, (IJRACSE), Volume 8 Issue 1, June 2022, Page 1-13.

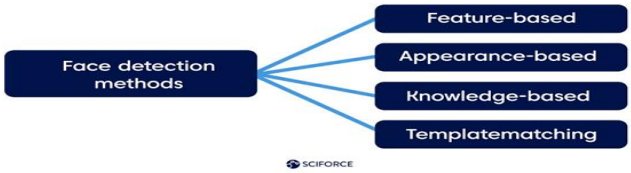
ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

1. Introduction

The use of software in society has increased widely in the last decades. As their exposure with humans increase, the interaction also has to become smoother and more natural. In order to achieve this, software have to be provided with a capability that let them understand the surrounding environment. Specially, the intentions of human being. When software is able to appreciate their surroundings, some sort of perception has been developed. Therefore, perception aims to mimic human senses in order to interact with their environment. Nowadays, computers have several ways to capture their environment state through cameras and sensors. Hence, using this information with suitable algorithms.

In the last years, the use of deep learning algorithms has been proven to be very successful as they are used mostly in the areas of pattern recognition and pattern classification problems. For instance, Jeremy Howard showed on his Brussels 2014 TEDx's talk how computers trained using deep learning techniques were able to achieve some amazing tasks. These tasks include the ability tolearn languages, to recognise objects in images. The facial expression of human emotion is one of the major topics in facial detection, and it can generate both technical and everyday applications. This projection constructs a system of deep learning model to classify a given image of human facial emotion into one of the several emotions. The tasks in the project include preprocessing the image data, augmentation to enlarge the existing dataset, test before training the model, training process, and prediction with evaluation.

The Development of FED has transformed from a simple passive system to an integrated intelligent system by adding advanced features. In this research, facial emotion detection in FED is used as a basis of decision making to produce automatic, effective and efficient integrated system. This research is done to give better performance than the traditional methods of image processing. This system is to gain information on facial recognition, feature extraction, emotion detection and music player.



1.1 Emotion Detection

Emotion detection is a classification technique which requires a pre-defined set of emotion classes and combines knowledge of artificial intelligence and psychology. Different emotion types are detected through the integration of information from facial expressions, body movement and gestures, and speech. The technology is said to contribute in the emergence of the so-called emotional or emotive Internet. Therefore the system can as the systems can adapt their responses and behavioural patterns according to the emotions of the humans and make the interaction more natural. Automatic emotion detection is an interesting research field which is used in several areas such as safety, health and in human machine interfaces. Researchers in this field are interested in developing techniques to interpret, code facial expressions

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

and extract these features in order to have a better prediction by system. With the remarkable success of deep learning, the different types of architectures of this technique are exploited to achieve. Thus we have undertaken five emotions to our implementation such as surprise, anger, happy, sad, neutral.

The three main components of Emotion Detection are as follows:

1. **Image processing** : It is the process of obtaining valuable information by conducting operations on an image, leading to an output of relevant components or features linked with the image.
2. **Feature Extraction** : The process of transforming the input data into the set of features is called feature extraction. Features define the behavior of an image, they show its place in terms of storage, efficiency in classification and in time consumption.
3. **Feature Classification** : A pattern recognition technique that is used to categorize a huge number of data into different classes

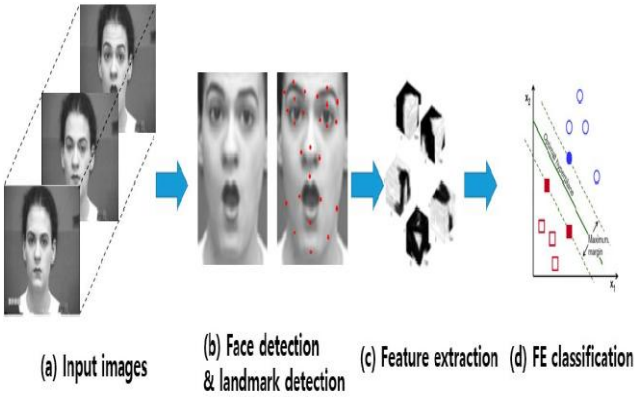


Figure 1: Emotion Detection

Sytem Design

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. The purpose of the System Design is to supplement the system architecture providing information and data useful and necessary for implementation of the system elements.

Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. A systemic approach is required for a coherent and well-running system. Bottom-Up or Top-Down approach is required to take into account all related variables of the system. A designer uses the modelling languages to express the information and knowledge in a structure of system that is defined by a consistent set of rules and definitions. The designs can be defined in graphical or textual modelling

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

languages.

Some of the examples of graphical modelling languages are

1. Unified Modelling Language (UML): To describe software both structurally and behaviourally with graphical notation.
2. Flowchart : A schematic or stepwise representation of an algorithm.
3. Business Process Modelling Notation (BPMN): Used for Process Modelling language. 4.Systems Modelling Language (SysML): Used for systems engineering.

1.2 DESIGN METHODS

1)Architectural design:To describes the views, models, behaviour, and structure of the system.2)Logical design:To represent the data flow, inputs and outputs of the system.
3) Physical design: defined as to how users add information to system and how data is modelled andhow data moves through the system

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the followingobjectives:
1.Convey information about past activities, current status or projections of the future.
2.Signal important events, opportunities, problems, or warnings.
3. Trigger an action.
4. Confirm an action.

MODELS

In this project, we have adopted the approach of using the model having the following specifications:

- Method: Deep Learning Model
- Type: Convolutional Neural Network
- Pre-trained on: FER2013,VGG16
- Category: Feature detection

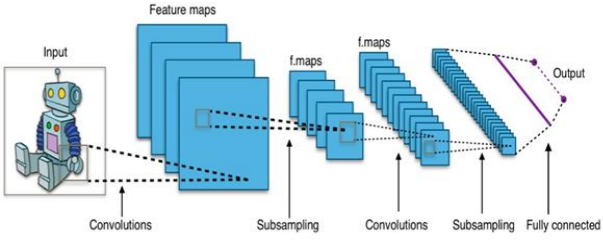


Figure 2: Model Architecture

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

1.3 VGG16 ARCHITECTURE

VGG16 is a convolution neural net (CNN) architecture. It is considered to be one of the excellent vision model architecture till date. VGG16 proved to be a significant milestone in the quest of mankind to make computers "see" the world. A lot of effort has been put into improving this ability under the discipline of Computer Vision (CV) for a number of decades. VGG16 is one of the significant innovations that paved the way for several innovations that followed in this field.GG16 keeps the data scientists and researchers worldwide interested despite the advent of many new and better scoring models since the time VGG was originally proposed. Here are afew use cases where you may find VGG16 practically in use.VGG stands for Visual Geometry Group is a standard deep Convolutional Neural Network architecture with multiple layers.The deep refers to the number of layers with VGG-16 consists of 16 convolutional layers.It is basedon the ground-breaking object recognition models.Developed as a deep neural network, the VGGNet also surpasses baselines on many tasks and datasets beyond ImageNet.The research paper has been published the model "Very Deep Convolutional Networks for Large-Scale Image Recognition".

1. Image Recognition or Classification – VGG16 can be used for disease diagnosis using medical imaging like x-ray or MRI. It can also be used in recognizing street signs from a moving vehicle.
2. Image Detection and Localization – We didn't discuss the detection abilities of VGG16 earlier, but it can perform really well in image detection use cases.
3. Image Embedding Vectors – After popping out the top output layer, the model can be used to train to create image embedding vectors which can be used for a problem like face verification using VGG16.

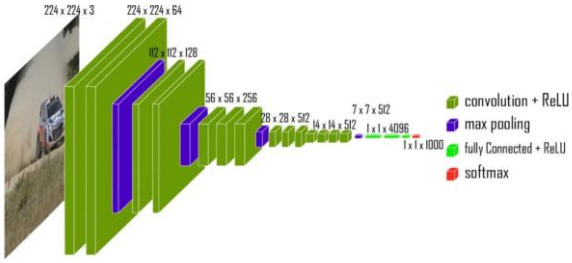


Figure 3. VGG16 Architecture

Classification and Implementation

A classifier is any algorithm that sorts data into labeled classes, or categories of information. A simple practical example are spam filters that scan incoming "raw" emails and classify them as either "spam" or "not-spam." Classifiers are a concrete implementation of pattern recognition in many forms of machine learning. Classifiers are where high-end machine theory meets practical application. These algorithms are more than a simple sorting device to organize, or "map" unlabeleddata instances into discrete classes. Classifiers have a specific set of dynamic rules, which includes an interpretation procedure to handle vague or unknown values, all tailored to the type of inputs being examined. Most classifiers also employ probability estimates that allow end

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

users to manipulate data classification with utility functions.

HAAR CASCADE CLASSIFIER

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle.

2. CODE FOR TRAINING

```
import numpy as np import tensorflow as tf
import cv2
import os
import matplotlib.pyplot as pltimport random
from tensorflow import keras
from tensorflow.keras import layers
from keras.utils.np_utils import to_categorical
from keras.layers import Dense, Conv2D,
Flatten, Dropout, MaxPooling2D, Activation
from keras.losses import
    categorical_crossentropy
    from keras.models import Sequential
    from keras.layers.convolutional import
    Convolution2D, MaxPooling2D,
    ZeroPadding2Dimport json
    import h5py

    Datadirectory = 'C:/Users/my
    Project/training_imgs/'Classes = [str(i) for i in
    range(5)]
    training_Data = []testing_Data = []

def create_training_Data(folder='train/'):
    new_path = os.path.join(Datadirectory, folder)
    for category in Classes:
    path = os.path.join(new_path, category)
    class_num = Classes.index(category)for img
    in os.listdir(path):

    try:
    new_array = cv2.imread(os.path.join(path,
    img))

if folder == 'train/':
    training_Data.append([new_array, class_num])
    else:
    testing_Data.append([new_array, class_num])

                                    except:
                                        pass

    create_training_Data()
    create_training_Data('test/')

    random.shuffle(training_Data)
    random.shuffle(testing_Data)

    X = [] #Training img Y = [] # Training class
```

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

```
x_test = [] #Testy_test = [] #Test

for features, label in training_Data:
    X.append(np.array(features, 'float32'))
    Y.append(label)

for features, label in testing_Data:
    x_test.append(np.array(features, 'float32'))
    y_test.append(label)

    num_features = 64
    num_labels = 5
    batch_size = 64
    epochs = 180
    width, height = 48, 48

    X = np.array(X, 'float32')
    Y = np.array(Y, 'float32')

    x_test = np.array(x_test, 'float32')y_test =
    np.array(y_test, 'float32')

    Y = to_categorical(Y, num_labels)
    y_test = to_categorical(y_test, num_labels)

    X /= 255.0
    x_test /= 255.0 model = Sequential()
    model.add(Conv2D(64, kernel_size=(2,3),
    activation='relu', input_shape=(X.shape[1:])))
    model.add(Conv2D(64, kernel_size=(2,3),
    activation='relu'))
    model.add(MaxPooling2D(pool_size=(2,2),
    strides=(2,2)))
    model.add(Dropout(0.5))

    model.add(Conv2D(64, (3, 3),
    activation='relu'))
    model.add(Conv2D(64, (3, 3),
    activation='relu'))
    model.add(MaxPooling2D(pool_size=(2,2),
    strides=(2,2)))model.add(Dropout(0.5))

    model.add(Conv2D(128, (3, 3),
    activation='relu'))
    model.add(Conv2D(128, (3, 3),
    activation='relu'))
    model.add(MaxPooling2D(pool_size=(2,2),
    strides=(2,2)))

    model.add(Flatten())

    model.add(Dense(1024, activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(1024, activation='relu'))
    model.add(Dropout(0.2))

    model.add(Dense(num_labels,
    activation='softmax'))model.summary()
    model.compile(loss =
    categorical_crossentropy, optimizer='sgd',
    metrics=['accuracy'])#sgd - Stochastic gradient
    descent optimizer


    model.fit(X, Y, batch_size=batch_size,
    epochs=epochs, verbose=1,
    validation_data=(x_test, y_test),shuffle=True).
```

2.1 CODE FOR FACE CAPTURE

```
import cv2

def getFace():
face_cascade =
cv2.CascadeClassifier(cv2.data.haarcascades
+'haarcascade_frontalface_default.xml')

vid = cv2.VideoCapture(0)img = prevGenre =
None
```

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

```
    while True:

    ret, frame = vid.read() cv2.imshow('Image',
    frame)
    gray = cv2.cvtColor(frame,
    cv2.COLOR_BGR2GRAY)faces =
    face_cascade.detectMultiScale(gray, 1.1, 4)
    found = False

if cv2.waitKey(1) & 0xFF == ord('q'):break
    for x, y, w, h in faces:

    roi_gray = gray[y:y+h, x:x+w] roi_color =
    frame[y:y+h, x:x+w]
    facess =
    face_cascade.detectMultiScale(roi_gray)

if len(facess): img = frame found = Truebreak
    if found:
    break

    return img
 if _name_ == '_main_':img = getFace()
    cv2.imshow('Image', img)
```

2.2 CODE FOR EMOTION DETECTION

```
    import os import cv2 import json import h5py
    import numpy as np
    from tensorflow.keras.models import
    model_from_json
    from tensorflow.keras.preprocessing.image
    import img_to_arraydef getEmotion(img):
    model = model_from_json(open('ferR.json',
    'r').read())model.load_weights('ferR.h5')

    face_cascade =
    cv2.CascadeClassifier(cv2.data.haarcascades
    +'haarcascade_frontalface_default.xml')

    #img = cv2.imread("C:\\Users\MY
    PC\OneDrive\Desktop\Pawan-Kalyan-
    1200by667.jpg")gray_img =
    cv2.cvtColor(img,
    cv2.COLOR_BGR2GRAY)
    faces =
    face_cascade.detectMultiScale(gray_img, 1.1,
    4)predictionImg = emo = None
    for x, y, w, h in faces:

    roi_gray = gray_img[y:y+h, x:x+w]roi_color
    = img[y:y+h, x:x+w]
    facess =
    face_cascade.detectMultiScale(roi_gray)if
    len(facess):
    for (ex, ey, ew, eh) in facess:

    face_roi = roi_color[ey:ey+eh, ex:ex+ew]
    final_image = cv2.resize(face_roi, (48, 48))
    final_image = np.expand_dims(final_image,
    axis=0)
    final_image = final_image/255.0

    prediction = model.predict(final_image)
    emotion_detection = ('angry', 'happy', 'neutral',
    'sad', 'surprise')predictionImg = prediction
    print(predictionImg)
    max_index = np.argmax(prediction[0]) emo =
    emotion_detection[max_index] return emo

    if _name_ == '_main_':
    pk = 'C:\\Users\MY
    PC\OneDrive\Desktop\Pawan-Kalyan-
    1200by667.jpg'mouni1 = r"C:\Users\MY
    PC\Downloads\neutral.jpeg"
    mouni2 = r"C:\Users\MY
```

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

```
PC\Downloads\WhatsApp Image 2022-05-23
at 12.05.51 PM.jpeg"
print(getEmotion(cv2.imread(pk)))
cv2.destroyAllWindows()
```

2.3 CODE FOR MUSIC PLAYER

```
from EmotionDetection import getEmotion
import os
import random
from playsound import playsoundfrom
pygame import mixer import keyboard
import cv2
from CaptureFace import getFace

paused = Falsemixer.init()

def Finish(): global paused

if paused or mixer.music.get_busy():return False
return True

def playSongs(): global paused img = getFace()
genre = getEmotion(img).capitalize()
print(genre)
path = 'C:\Final year project\Songs\\' + genre
songs_count = 0
allSongs = set()

for dirs, files, songs in os.walk(path):for
each_song in songs:
allSongs.add(each_song)songs_count += 1

allSongs = list(allSongs)
random.shuffle(allSongs)

print("Pause : 'p'")
print("Resume : 'r'")
print("Next : 'm'")
print("Prev : 'n'")
print("Exit : 'q'")

exitLoop = Falsesong = 0

while len(allSongs): curr_song = allSongs[song]
playSong = str(path + '\\' + curr_song)
mixer.music.load(playSong)
mixer.music.set_volume(5)
mixer.music.play()
while True and not Finish(): if
keyboard.is_pressed('q'):
try:
mixer.music.stop()except:
pass exitLoop = Truebreak

elif keyboard.is_pressed('r'):
mixer.music.unpause() paused = False
elif keyboard.is_pressed('p'):mixer.music.pause()
aused = True

elif keyboard.is_pressed('m'):
song = (song + 1) % len(allSongs)
mixer.music.stop() curr_song =
allSongs[song]
playSong = str(path + '\\' + curr_song)
mixer.music.load(playSong)
mixer.music.play()

elif keyboard.is_pressed('n'):mixer.music.stop()
song -= 1

if song < 0:
song = len(allSongs)-1

curr_song = allSongs[song]
playSong = str(path + '\\' + curr_song)
mixer.music.load(playSong)
mixer.music.play()
```

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

```
if exitLoop:break

    song = (song + 1)%len(allSongs)
cv2.destroyAllWindows()
if _name_ == '_main_':playSongs()
```

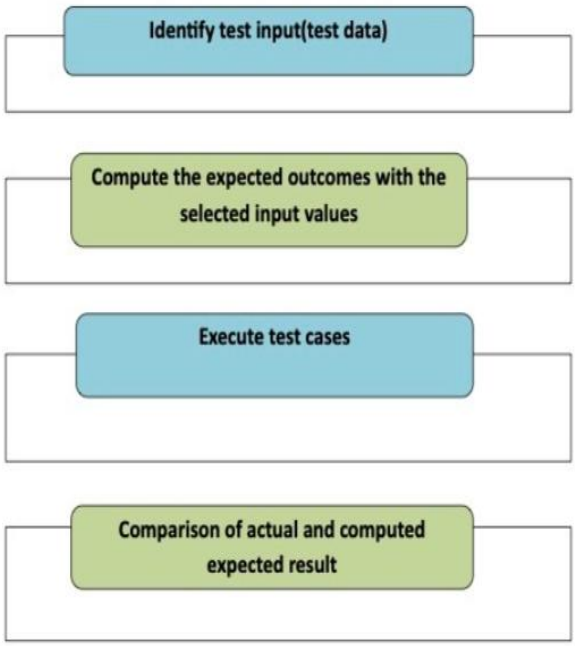
Functional tests provide systematic demonstrations that functions = tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

1. Valid Input : identified classes of valid input must be accepted.
2. Invalid Input : identified classes of invalid input must be rejected.
3. Functions : identified functions must be exercised.
4. Output : identified classes of application outputs must be exercised.

Following is a step by step process on **How to do Functional Testing** :

1. Understand the Functional Requirements
2. Identify test input or test data based on requirements
3. Compute the expected outcomes with selected test input values
4. Execute test cases
5. Compare actual and computed expected results



3. Output Screenshots

A performance comparison of the various architectures and approaches used in the literature is presented in the table below. Please note that all our experiments use the Haar cascade classifier during pre-processing.

The best accuracies achieved by our respective models are as follows:

For our project we have taken training data and test data with the 30000 parameters as input where the test data is compared with the training data with the class of five labels of emotions such as happy, angry, sad, surprise, neutral.

In the output screenshots the comparison is shown between the baseline model and our final model which shows the difference in between the accuracy and epochs that both of the models produce as output.

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

MODEL/APPROACH	ACCURACY
3D CONVOLUTIONAL	3D CONVOLUTIONAL
I3D, Carreira and Zisserman [19]	75.4%
Kinetics pre-trained, Wijnandset al. [20]	73.9%
2D CONVOLUTIONAL	2D CONVOLUTIONAL
InceptionV1, Szegedy et al. [18]	69.6%
OURS(FINAL MODEL)	71 %

Moreover, we also visualize the training and testing accuracy as well as loss after each iteration.The concatenated graphs of the same have been shown below:

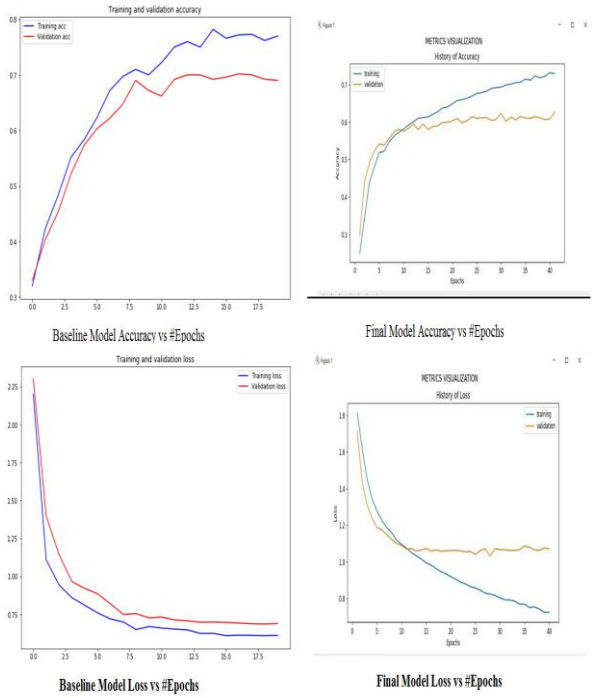


Fig. 4 visualize the training and testing accuracy as well as loss after each iteration concatenated graphs

Fig 5. Emotion Input 1: Happy, Emotion Input 2: Sad, Emotion Input 3: Neutral, Emotion Input 4: Surprise, and Emotion Input 5: Angry

4. Conclusion

Emotion recognition using facial expressions is one of the important topics of research and has gathered much attention in the past. It can be seen that the problem of emotion recognition with the help of image processing algorithms has been increasing day by day. Researchers are continuously working on ways to resolve this by the use of different

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

kinds of features and image processing methods. The applications of image processing algorithms in the field of both medical science.

By this, we have proposed a deep learning based facial emotion detection method from the image.

Future Enhancement

Emotion recognition has gained a lot of importance in all aspects of life and if a robust algorithm implemented which can accurately classify the emotions of the person, then a great deal of advancement in the industry can be achieved with the help of this. The system has successfully been able to capture the emotion of a user. It has been tested in a real-time environment for this predicate. A simple system is proposed here for the music recommendation using face emotion recognition. It suggests music by extracting different facial emotions of a person: Happy, anger, surprise, neutral. There is a degree for further upgrades and enhancements. Progressively effective approaches to incorporate different highlights and functionalities should, in any case, be investigated due to the lopsided nature of each element set. It is additionally seen that to improve the exactness of the arrangement framework the informational collection used to construct the grouping model could be expanded further.

This software can also be embedded into android applications and we can create an interface like leading music apps like Spotify, Jio saavn etc.we can incorporate this software in such kind lo apps and get more user experience by just facial recognition.This enhancement can lead to more improvement in this project.

5. References

1. E. Correa, A. Jonker, M. Ozo and R. Stolk, "Emotion recognition using deep convolutional neural networks",ELSEVIER, 2016.

2. Y. I. Tian, T. Kanade and J. F. Cohn, "Recognizing action units for facial expression analysis", IEEE, 2001.

3. Ramya Ramanathan, Radha Kumaran, Ram Rohan R, Rajat Gupta, and Vishalakshi Prabhu, “An intelligent music player based on emotion recognition”, 2nd IEEE International Conference on Computational Systems and Information Technology for Sustainable Solutions,2017.

4. Ahlam Alrihail, Alaa Alsaedi, Kholood Albalawi, Liyakathunisa Syed, “Music recommender system for users based on emotion detection through facial features”,JIEEE,2019.

5. Preema J.S, Rajashree, Sahana M, Savitri H,” Review on facial expression-based music player”, International Journal of Engineering Research & Technology (IJERT), 2018.

6. AYUSH Guidel, Birat Sapkota, Krishna Sapkota, “Music recommendation by facial analysis”,ResearchGate, 2020.

7. Ahmed Hamdy AlDeeb, “Emotion- Based Music Player Emotion Detection from Live Camera”, ResearchGate, 2019.

8. Ch.sadhvika, Gutta.Abigna, P.Srinivas reddy, “Emotion-based music recommendation system”, International Journal of Emerging Technologies and Innovative Research (JETIR), 2020.

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

9. Mirim Lee and Jun-Dong Cho, "Logmusic: context-based social music recommendation service on mobile device", Ubicomp'14 Adjunct, 2014.
10. D. Gossi and M.H. Gunes, "Lyric-based music recommendation" in Studies in computational intelligence, Springer Nature, 2016.