Emotion Recognition from Facial Expression of Autism Spectrum Disordered Children using Image Processing and Machine Learning Algorithms

Inspiring Excellence

SUBMISSION DATE: 22.07.18

SUBMITTEDBY:

Nishat Tasnim (14301022)

AtiyaKhwaja (14101103)

Humayra Rashid (14101099)

Department of Computer Science and Engineering

Supervisor:

AmitabhaChakrabarty, Ph.D

Assistant Professor

Department of Computer Science and Engineering

Declaration

We, hereby declare that this thesis is based on results we have found ourselves. Materials of work from researchers conducted by others are mentioned in references.

Signature of Supervisor **Signature of Authors**

AmitabhaChakrabarty, Ph.D
Assistant Professor
Department of Computer Science and Engineering
BRAC University

Name (ID)

Name(ID)

Name(ID)

Md. Saiful Islam
Lecturer
Department of Computer Science and Engineering
BRAC University

ABSTRACT

Emotion recognition from facial expression is one of the most popular research sectors in Computer Science for last several years. As face recognition of different people is already established and become very common in almost everywhere, emotion recognition from these faces can be a new feature in the existing processes. On the other hand, image processing is the new emerging field in Computer Science. There are already several well renowned researches on facial expression recognition using image processing. In our research, we have recognized the emotions of children who have Autism Spectrum Disorder (ASD).Most autistic children show symptoms of withdrawal from social interactions and a lack of emotional empathy towards others. Individuals with autism exhibit difficulties in various aspects of facial perception, such as facial identity recognition or recognition of facial expressions. Our goal is to find a way that can understand their emotions accurately so that further improvement based on our research in this field can find a pattern of their expressions for different emotions and can make it easy for them as well as for the people surrounded by them to interact with each other. For our research, we have used seven Machine Learning algorithms along with Convolutional Neural Networks (CNN) through the datasets of different expression of ASD children and compared the accuracy to find out the algorithm/s that identifies their expression most perfectly. We also demonstrated two feature extraction techniques that are PCA and LBP to improve the accuracy of the algorithms. Our approach shows some mentionable output. The best result we get is after using SVM with PCA feature extraction that gives almost 68.2% of accuracy.

Keywords: Facial Expression Recognition (FER), Image Processing, Autism Spectrum Disorder (ASD), Machine Learning Algorithms, Convolutional Neural Network (CNN), Feature Extraction Techniques.

Acknowledgement

First person whom we should convey our heartfelt gratitude is none other than our respected thesis supervisor Dr. AmitabhaChakrabarty for his continuous support and inspiration. He was immensely helpful throughout our whole research time. After that, it should also be mentioned the name of the Co-supervisor, Md. Saiful Islam as we are very much grateful for his continuous suggestion to improvise our work. Both respectively supported and helped us throughout this journey. Lastly, the name that must be mentioned is of our Research Assistant, TasniaAshrafiHeya, who was also in continuous touch with us and helped us accordingly whenever we sought help to her.

In another note, we are also very much thankful to the deputy director of Society for the Welfare of Autistic Children (SWAC), Mr. Mofijul Islam, for letting us collect the data from his institution.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
Introduction

Image processing has multiple scopes to make our life easier. From pattern and texture recognition to facial expression recognition, image processing can be used in natural language processing or even in biometric analysis. From retrieving information to identify criminals, image processing can make our life easier by contributing also in medical science. For understanding human behavior and expression, facial expression recognition by image processing is currently one of the rising research fields of Computer Science. There are several well renowned prior works in facial expression recognition using image processing. So, we want to choose a field which is less explored and where we can offer some contribution. ASD(Autism Spectrum Disorder) is viewed as a comprehensive neural developmental disorder that affects social interaction, language or behavioral skills of children.ASD children has difficulties to express their feelings and it is less explored field even by the medical science researchers. ASD children have some types of facial expressions that differ from normal people. So, we decided to take this less explored challenging field as our research project to make some contribution. Moreover, determining the integrity of facial emotion recognition (FER) in ASD is important to our theoretical understanding of autism.

1.1 Motivation

With the recent development in machine learning area, it has shown some increasing accuracy rate as well as optimized performance. If we consider the image processing sector, the research field is even more demanding. As we wanted to do our research in a field which is currently expanding, we decided that image processing will be our right choice. As our topic is sensitive and confidential and also there were some ethical obligations, there was no secondary datasets available from any sources. However, we needed a huge number of images of ASD children to perform our research

properly, so we decided to make our own datasets. We were ready to take this challenge and accomplished it. This also motivates us to do our research properly as we also felt the social liability to do some good contribution to be a better citizen.

1.2 Objectives

Our main objective is to find out if the conventional and popular machine learning algorithms along with the popular image processing algorithm CNN can keep their performance as good as they perform for emotion recognition of non-ASD people. Also, our target is to figure it out which algorithm can perform the best with the highest accuracy as well as which emotions of ASD children are easily recognized. There are many therapies for them to improve their expressions and behavior, but our proposal is based on their initial expression. As they cannot express according to their needs, so we will try to find a way so that people around them will be able to understand their needs more accurately and be more empathetic towards them. So, we look into generate facial expression recognition on low-resolution images and compare and contrast it with other learning models. So that we can reach a conclusion where less computational power can be used to achieve high level of image processing. For fulfilling our target, we followed the footsteps of our prior researchers who worked in the same field and tried to find out if the algorithm goes the same with the ASD children's images.

CHAPTER 2

Literature Review

Background Study

In recent years, researcher have made considerable progress in developing emotion recognition for ASD children using Eye gaze, polyvagal theory, RSA, FEFA[6,7]. Understanding emotions usually requires multi-sensory processing for example Correlations were calculated to examine the relationship between averaged fixation duration percentage for three regions (EYE, MOUTH, OFF) within each emotion and averaged recognition latency and sum of errors for each emotion[6].Some researchers used speech prosody, which is a commanding expression has been reported to be the easiest to recognize, while the sad expression is the most difficult in all basic emotion for example (Happy, Angry, Sad, Neutral).so, most of processing is subconscious and it requires the ability to divide attention and the focus gaze on relevant information[7]. There are several work on emotion detection using image processing but most of the work on the Normal people [2,3,4,5]. Facial recognition using deep learning, more specifically using Convolutional Neural Network or CNN has established itself as the most efficient model giving the best results on image classification problems. Some well-known researchers use CNN approach where they have used dropout and data augmentation to solve the problem of over fitting. One mentionable CNN approach was by LeCun in LeNet-5 system. Such standard CNN layers are structured using multiple layers, on each layer groups are from based on the features they work on and these groups feed forward to the next layer and at the end it connects with the fully connected layer[3,4,5].A comparative analysis of prior image processing techniques using machine learning has already been done which has denoted comparison of efficiency of machine learning models on facial expression detection. They have implemented two shallow learning models and one deep learning model for the purpose of comparison[1,2].Inception architecture reduce computational resource usage in highly accurate image classification using deep learning and it seems quite efficient. Also those behavioral studies of FER in ASD have mixed results so we will be addressing demographic and experiment related factors discussing the possibility that compensatory mechanisms might enable some individuals with ASD[5,7]. Evidence for such mechanisms comes in part from eye-tracking, electro psychological studies which often show

abnormal eye gaze patterns in emotion processing circuitry in ASD. We searched about a formal meta-analytic approach for comparing purpose along with Web of Science, Psych INFO and PubMed.[8,9].. Also we have learned that most of the previous work has used FACS as a frame-work for classification. In addition to this, previous studies have traditionally taken two approaches to emotion classification according to Fasel and Luettin - one is a judgment based approach and another one is sign-based approach[9,10].The judgment approach develops the categories of emotion in advance such as the traditional six universal emotions. The sign-based approach uses a Facial Action Coding System (FACS) system, encoding action Units in order to categorize an expression based on its characteristics.*NOTE*: Any paper meeting the above requirements will be acceptable, but the same paper must be used throughout to avoid variations in color and texture[10,11,13].

CHAPTER 3

Data Set Analysis

3.1 Data Collection

For our research purpose we have built our own data set. We have searched for secondary datasets in some popular data repositories like Kaggle, UCI etc. But as the datasets we needed were the facial images of ASD children, the data was quite confidential and there were some ethical obligations to publish the data in open sources for public access. So after searching every nook and corner we failed to get any secondary data. Then we decided to make our own primary dataset. But still there was so many obligations and confidentiality that we need to go through. Data collection became the most challenging part of our research. After several discussions between our teammates and the supervisor, we finally decided to take the challenge all by ourselves.

The journey of data collection was not very smooth though. We have to mention here that our respected supervisor, Dr. AmitabhaChakrabarty sir was so much co-operative throughout the whole time. We have discussed with him the difficulties that we are being faced for data collection. Our supervisor asked us to make an application form for the confidentiality purpose. We made a code of conduct that we will strictly follow and our supervisor also signed it on behalf of us. That was a big step for make it easier for us. The all three members of our team then contacted with three different institutions that work with children with Autism Spectrum Disorder. As these are not a common type of schools, the institutions are not always well known by everybody and also they are situated in the peripheral regions of the city. We were also struggling with finding and contacting with such institutions. As we asked our fellow friends, relatives about any link with such institutions, they also helped us a lot to find out and made a link with the institutions so that we could use their reference. Lastly we short listed three institutions. One of them was a local institution with few students; other two were Society for the Welfare of Autistic Children (SWAC) and Autistic Welfare Foundation (AWF). The next part was to make contact with those institutions. Three of us separately found links with three separate institutions; we mailed the chairperson of those institutions and waited for a suitable time they allocate for us. As we needed the natural expressions of them and didn't want to manipulate their expressions, we needed to spend a whole day with them to collect the accurate data of their

expressions. In the allocated day we also had to gone through a short interview for verification before we finally get permission to take images.

Three of us spend a whole day in each institution. So we actually spend whole three days to three different institutions. From the three institutions we got all kind of help and among them we were able to take pictures from two institutions. From all the institutions, we could take facial pictures of total 30 children. The age range was between 5years to 15years. Among the students, 20 were male and the rest 10 were female. As we didn't want to hamper their natural behavior, we spend the whole day with them with full-time monitoring. We mainly selected the troubled children who usually have frequent mood swings within a day. We captured a lot of images of them. While taking the pictures, we had a brief discussion with that individual's instructor about the emotions of them according to the expression they were making. We labeled the dataset accordingly. As they were not stable and steady, many of the images came out blurry which we couldn't use as our dataset. So later we gone through a selection process by which we deducted the images that was not so clear or the expressions was confusing. After all the selection, we were able to retrieve approx. 2000 images as our final dataset.

3.2 Data Labeling and Refining

As we said earlier, we collected this image data from three different ASD Institution . Machine learning are classified into two broad categories Supervised learning and Unsupervised learning. We choose supervised learning categories for that reason we have made training data and the test data. We tried to identify four categories of emotion for example: Happy, Anger, Sad and Neutral from ASD children's facial expression. Following this categories, We have labeled the data with the help of the instructor of that institutions. As we know that it is very hard to understand the emotions of ASD children. Understanding emotions usually requires multiple elements for analysis, for example their facial gestures, their behavior. Those instructors are working with them for long time therefore, whenever we collected the image, we asked them about the ASD children's mood and we labeled our data. Our dataset is in JPG format. We resize the image into 2500 (image height 50 x image width 50) pixels and put them into gray scale. The whole trained data put into an array and shuffled the images so that, only same categories data would not remain in validation set.

3.2.1 Training Dataset

The actual data set that we label and used for train the model. The model sees and learns from the model. We used 1600 data for training from the whole dataset, which is 80%.

3.2.2 Validation Dataset

While tuning model hyper parameters, the sample data used to provide an unbiased evaluation of a model fit on the training dataset [8].Among the 80% training data, we used 70% of them for creating the model and remaining 30% is used to fit the model. Once the model is well -fitted then the model is considered as trained data.

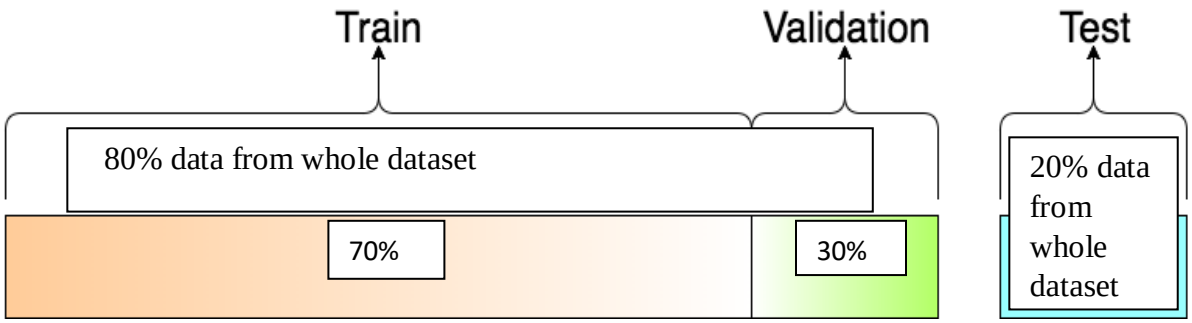

Figure 3.2.1: A visualization of the splits[8].

We split the train data into 70% and 30% for creating the unbiased model. After creating the model neither for emotion detection we tested some data, which was, nor in the training part.

3.3 Test Dataset

To evaluate the model the test dataset provides us the standard. When our model is completely trained, the test dataset is used. We got different type of accuracy from different type of algorithms[8].we used 400 data(which is 20% data in whole data set) for testing. Which was not in training part.

CHAPTER 4

System Overview

4.1Convolutional Neural Network (CNN)

It is a class of deep and a type of feed-forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex. Convolutional Neural Networks are the art of the model architecture for image classification and emotion recognition. For the classification CNN apply a series of filtering to the raw pixel data of an image to extract and learn higher-level features. CNN contains three layers.

- **Convolutional Layer**:

  Here the image is filtered to every possible position and then multiplied the corresponding pixel values. Then adding them and divide by the total number of pixels in the features. This layer then typically applies a ReLU activation function to the output to introduce nonlinearities into the model [1].

- **Pooling Layers:**

  Here we down sample the image data which already extracted by the convolutional layers so that we can reduce the dimensionality of the features in order to decrease processing time. We used Max_pooling2d () the algorithm which extracts [1].

- **Dense(Fully connected) Layers:**

  The features extracted from the convolutional layers and downsampled by the pooling layers are classified in this layer. In a dense layer, every node in the layer is connected to every node in the preceding layer. The final dense layer in a CNN contains a single node for each target class in the model with a softmax activation function to generate a value between 0–1 for each node (the sum of all these softmax values is equal to 1). We can interpret the softmax values for a given image as relative measurements of how likely it is that the image falls into each target class.

CNN procedure:

Here, we are explaining how we have implemented the model to classify the emotions of the Autistic children.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 50, 50, 1)         0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 46, 46, 64)        1664
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 23, 23, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 19, 19, 128)       204928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 9, 9, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 10368)             0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               2654464
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 32)                8224
_________________________________________________________________
dropout_2 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_3 (Dense)              (None, 4)                 132
=================================================================
Total params: 2,869,412
Trainable params: 2,869,412
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 4.1: Layers of CNN Model.

4.1.1 Input Layer:

For creating convolutional and pooling layers for two dimensional image data this method is used.

- **Batch_Size:** we used around 2000 examples during performing gradient descent during training.
- **Image_height:** The height of the example image is 50.
- **Image_Width:** The weidth of the example image is 50
- **Channels:** Number of color channels in the example image is 1, as we used monochrome images.

- **Data_Format:** A String, the desired shape for our input layer is **[batch_size, 50,50,1].[1]**

4.1.2 Convolutional Layer 1:

In our 1st convolutional layer , we applied 32 5x5 filters to the input layer, with a ReLU activation function . here conv2d() method also used in the layers module to create this layer. For example:- [convent, 32, 5, activation="relu"].

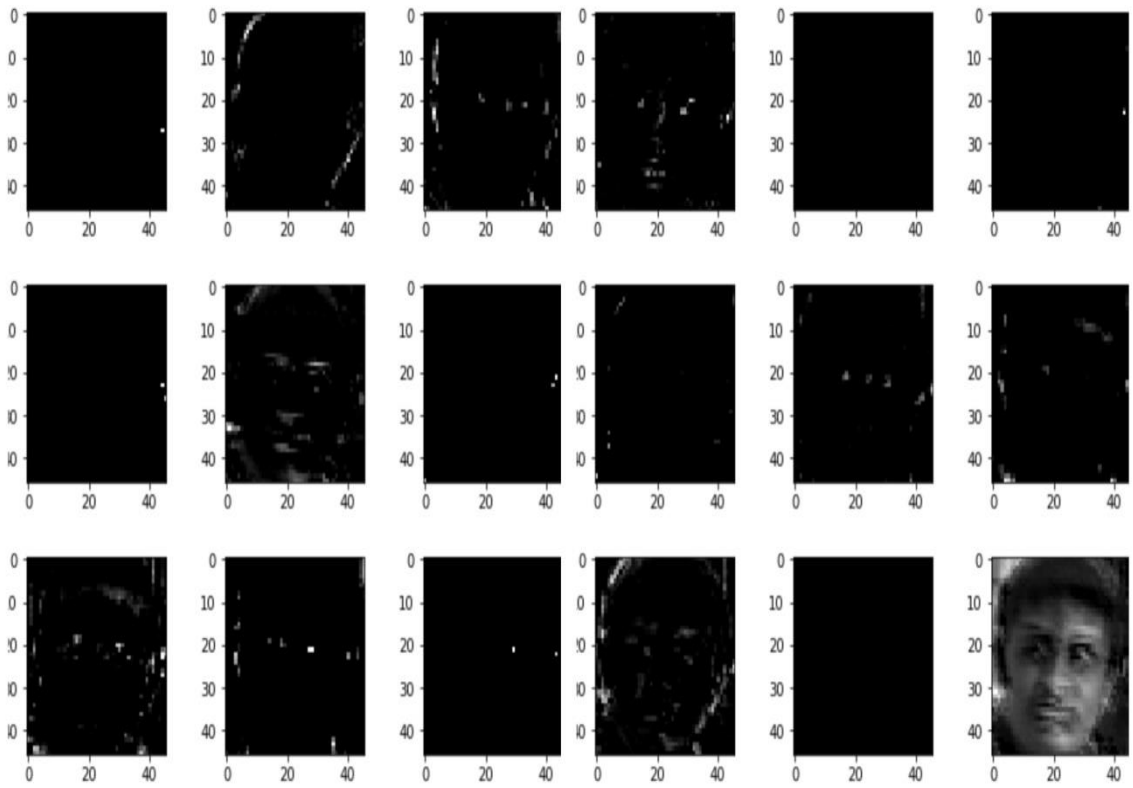

Figure 4.2: The image overview after convolution layer 1.

The padding argument specifies one of two enumerated values valid (default value) or same. To specify that the output tensor should have the same height and width values as the input tensor, we set padding=same here, which instructs TensorFlow to add 0 values to the edges of the input tensor to preserve height and width of 50.[1]

Pooling Layer 1:

Now, we have connected our first pooling layer to the convolutional layer already we just created. Max_pooling2d ()method construct a layer that perform max pooling with a 5x5 filter and stride of 2.for example :-**[Max_Pool_2d(convent,5)].**

The strides argument specifies the size of the stride. Here, we set a stride of 2, which indicates that the sub-regions extracted by the filter should be separated by 2pixels in both the height and width dimensions (for a 5x5 filter, this means that none of the regions extracted will overlap).[1]

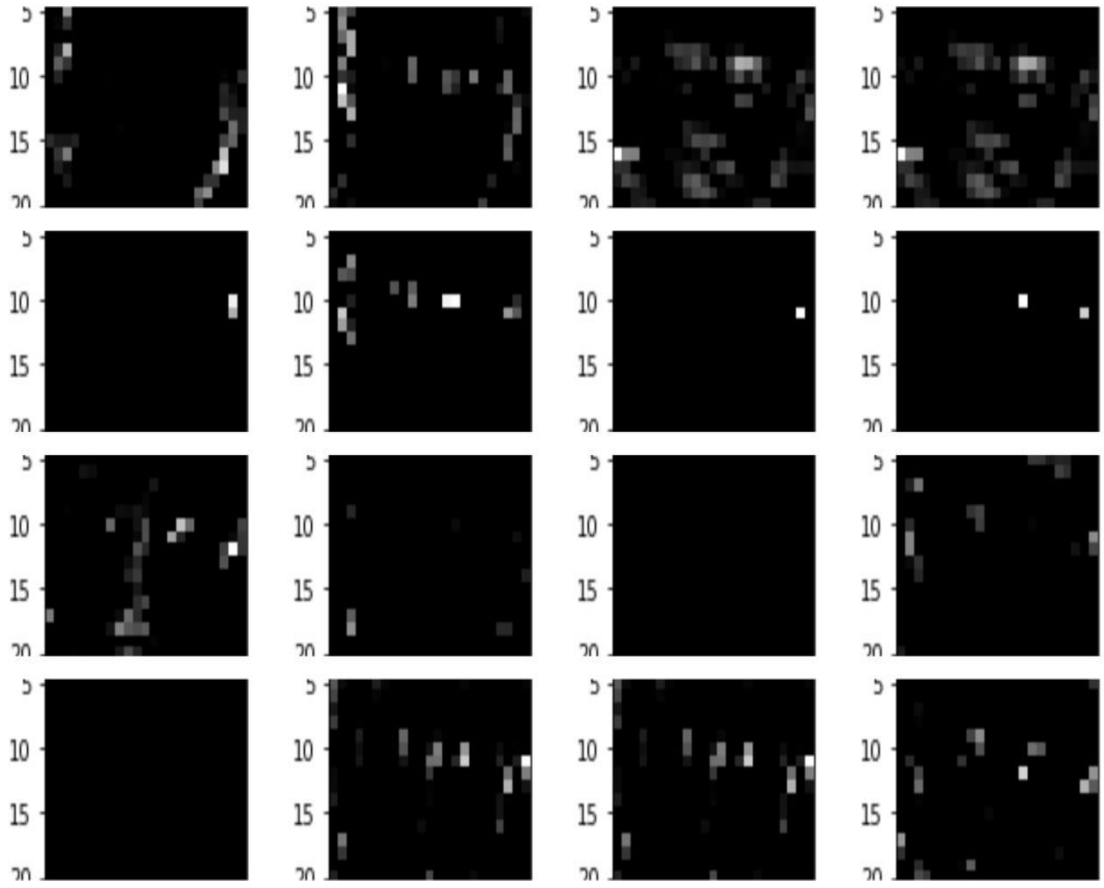

Figure 4.3: : The image overview after pooling layer 1

Convolutional Layer 2:

We used a second convolutional and pooling layer to our CNN using `conv2d()` and `max_pooling2d()` as before. For convolutional layer #2, we configure 64 5x5 filters with

ReLU activation. The tensor `conv2` as output. `conv2` has a shape of `[`*batch_size*`, 50, 50, 64]`, the same height and width as `pool1` (due to `padding="same"`), and 64 channels for the 64 filters applied.

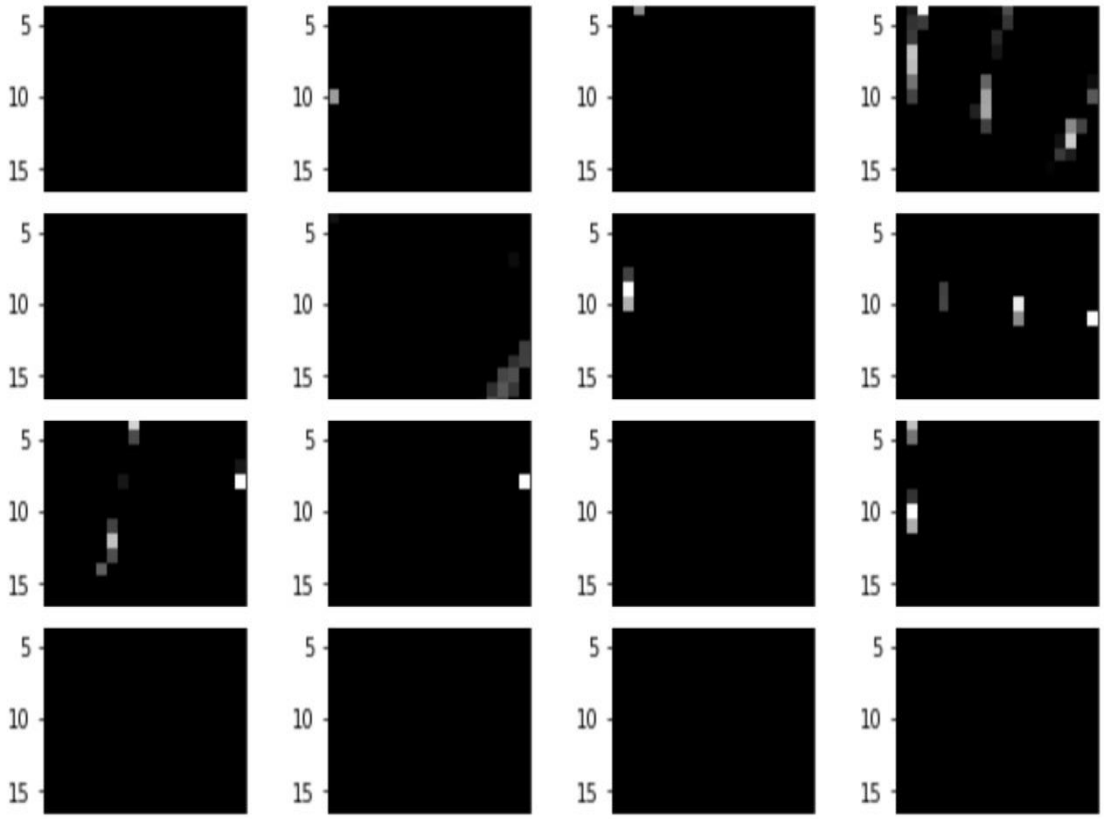

Figure 4.4: After Convolutional layer 2

Pooling Layer 2:

For pooling layer #2, we use the same specs as pooling layer #1 (a 5x5 max pooling filter with stride of 2). Pooling layer #2 takes `conv2` as input, producing `pool2` as output. `pool2` has shape `[`*batch_size*`, 50, 50, 64]` (50% reduction of height and width from `conv2`).

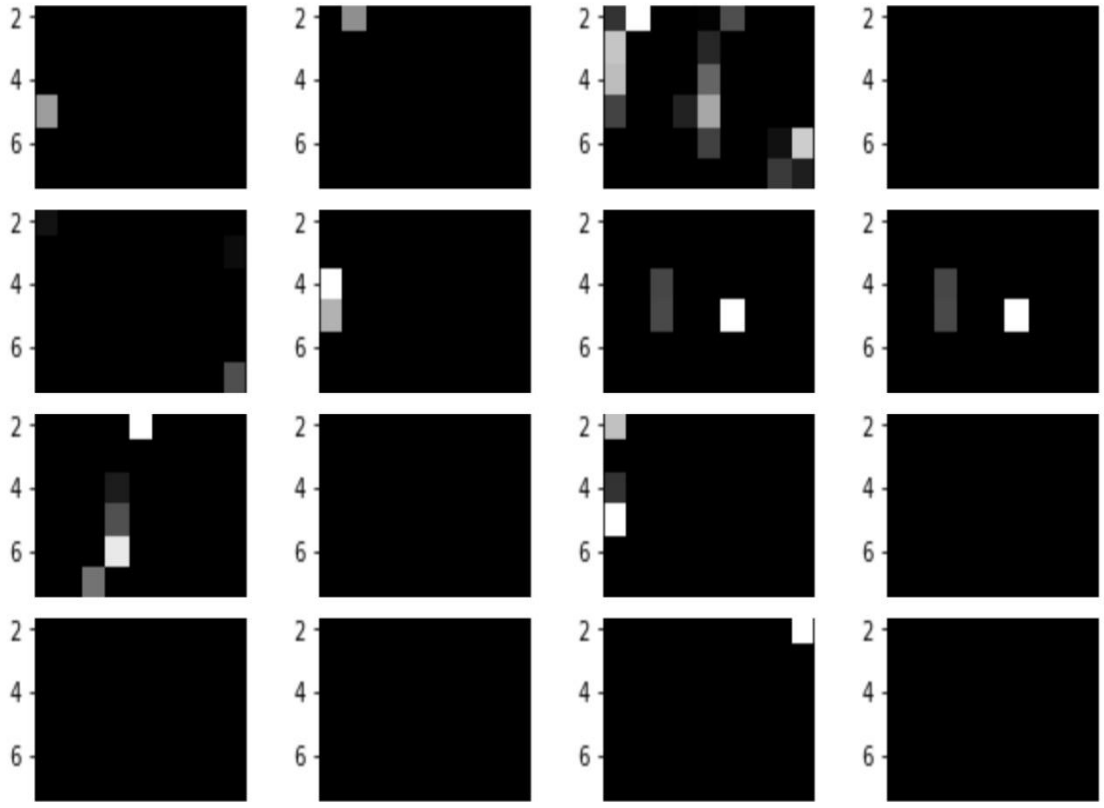

Figure 4.5: After Pooling layer 2

Flatten layers:

After some convolutional layers we need to make sure that fully connected layer for that reason we need the flatten layers. Convolutional layers have local limitation but fully connected layers don't have(which only observe some local part of an image by using convolutional filters). We can combine all the previous local features of convolutional layers. Each feature map channel in the output of a CNN layer is a "flattened" 2D array created by adding the results of multiple 2D kernels.[9]

```
length of flatten_1  layer  (layer 5): 10368
layer 5 : [0. 0. 0. ... 0. 0. 0.]
```

Dense Layerand Dropout layer: we take our filtered and shrined images and put them into a single list. Here we reshape our data. For example: [data. Reshape(1, 50 x50)]

Table 4.1: Dense layer values 1

```
                length of desnse_1  layer (layer 6) : 256
 (layer 6) : [0.0000000e+00 2.3729305e+00 3.7748404e+00 2.2077508e+00
                            0.0000000e+00
 0.0000000e+00 2.6354992e+00 0.0000000e+00 0.0000000e+00 4.3587561e+00
 2.2538927e+00 2.9638777e+00 0.0000000e+00 0.0000000e+00 1.5309749e+00
 0.0000000e+00 0.0000000e+00 8.3192670e-01 0.0000000e+00 0.0000000e+00
 3.9233305e+00 2.9433949e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 6.1321125e+00 0.0000000e+00 4.8696607e-01
 0.0000000e+00 4.6092372e+00 0.0000000e+00 1.1384717e+01 5.9463823e-01
 0.0000000e+00 0.0000000e+00 1.6646491e+00 0.0000000e+00 5.1324205e+00
 2.8973572e+00 1.6894200e-01 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 9.9143988e-01 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 2.1483757e+00 3.3454871e+00 0.0000000e+00 0.0000000e+00 1.7124316e+00
 0.0000000e+00 6.9773436e-01 7.9106493e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 1.4106004e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 2.8596401e+00 0.0000000e+00 2.7739009e-01 2.5756853e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 5.6175532e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 8.7539154e-01 0.0000000e+00 5.8950987e+00 2.6700025e+00 3.4302168e+00
 1.7767882e+00 5.4341882e-02 1.7033203e-01 0.0000000e+00 0.0000000e+00
 3.9933879e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 2.3552923e-01 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 2.8506989e+00 1.4393860e+00 3.0905590e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 1.8731205e+00 0.0000000e+00
 0.0000000e+00 7.4747187e-01 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 8.3462477e-01 0.0000000e+00 1.5464497e+00 4.3496752e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 3.7728231e+00 7.6642518e+00 0.0000000e+00 1.4023080e+00
 0.0000000e+00 6.6744900e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 1.6441588e+00 0.0000000e+00 0.0000000e+00 3.8656049e+00
 6.5477409e+00 0.0000000e+00 3.8757899e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 2.4596231e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 7.0835811e-01
 1.8160945e+00 0.0000000e+00 4.9030919e+00 1.2802646e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 1.0236928e+00
 4.1935158e+00 5.9561625e+00 0.0000000e+00 2.6688781e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 5.2300229e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 3.6337681e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 2.7733212e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 4.1200385e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 2.1603072e+00
                            8.2658224e+00]
```

Table 4.2: Values of Dropout Layer 1

```
length of dropout_1 (layer 7) : 256
(layer 7) : [ 0.          0.          0.          0.          0.
0.
  3.764999    0.          0.          6.2267947   3.2198467   0.
  0.          0.          2.187107    0.          0.          1.1884668
  0.          0.          5.604758    4.2048497   0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          8.76016     0.          0.69566584  0.
  6.584625    0.         16.263882    0.8494832   0.          0.
  0.          0.          7.3320293   4.139082    0.24134572  0.
  0.          0.          0.          0.          0.          1.4163427
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          3.0691082
  4.7792673   0.          0.          2.4463308   0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.39627156  0.
  0.          0.          0.          0.          0.          8.025076
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          1.2505593   0.          0.
  0.          4.9003096   2.538269    0.07763126  0.24333148  0.
  0.          5.70484     0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          2.0562658   4.4150844   0.          0.
  0.          2.6758866   0.          0.          1.067817    0.
  0.          0.          0.          0.          0.          0.
  0.          0.          1.1923211   0.          2.2092137   6.213822
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          5.389747
 10.948932    0.          2.003297    0.          9.534986    0.
  0.          0.          0.          0.          0.          0.
  5.5222926   9.353916    0.          5.536843    0.          0.
  0.          3.5137475   0.          0.          0.          0.
  0.          0.          0.          1.0119401   2.5944207   0.
  7.004417    1.8289495   0.          0.          0.          0.
  0.          1.4624183   5.990737    8.508803    0.          3.812683
  0.          0.          0.          0.          0.          0.
  0.          0.          7.4714613   0.          0.          0.
  0.          0.          5.17798     2.0548513   0.          5.073654
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          5.1910973   0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          3.0861533  11.808318  ]
```

Table 4.3: values of dense layer 2

```
length of desnse_2 layer (layer 8) : 32
(layer 8) : [ 0.         0.         6.320149   0.         3.8272958  0.
  0.         0.8701182  5.033115   0.         4.9604425  0.
  0.         3.7448714  0.        10.353118   0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         5.9919477]
```

Table 4.4: values of dropout layer 2

```
length of dropout_2  layer (layer 9) : 32
(layer 9) : [ 0.         0.         6.320149   0.         3.8272958  0.
  0.         0.8701182  5.033115   0.         4.9604425  0.
  0.         3.7448714  0.        10.353118   0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         5.9919477]
```

Table 4.5: values of Dense layer 3

```
length of desnse_3  layer (layer 10) : 4
(layer 10) : [2.3522315e-04 6.9591275e-05 9.9403149e-01 5.6636799e-03]
```

We put this reshape data input and activate the ReLU activation . Took the number of neurons and activation function as arguments. We used Dropoutfor dropping out the hidden and visible units and averaging the model efficiently.

Flow chart:

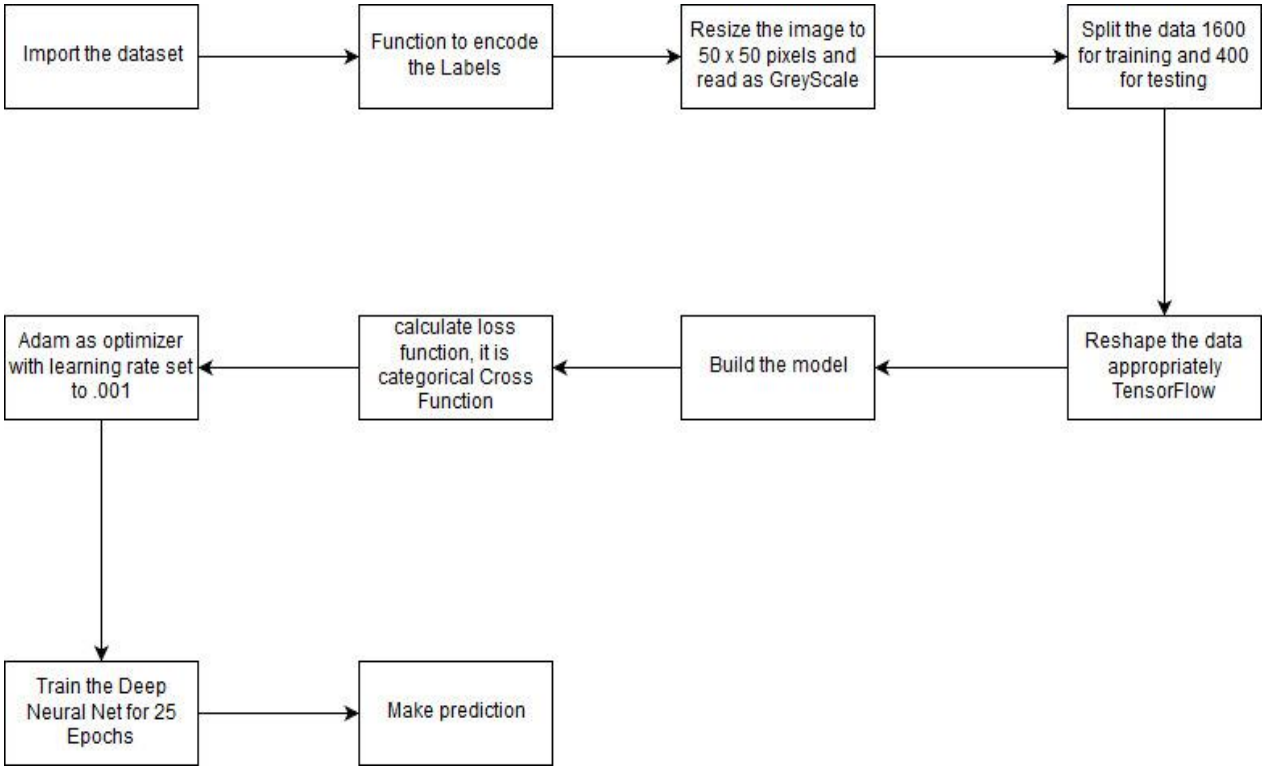

Figure 4.1.6: The flow char of implementing the CNN

4.2 Logistic Regression

Logistic regression, in other word logit model, is the appropriate regression analysis to conduct when the dependent variable is dichotomous or binary. Like all regression analysis, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. As it is used to predict a dependent variable which can assume more than 2 values, in this case, the model will be called "multinomial logistic regression". In multi class case, there will be set multiple-class to it and for this, the algorithm uses cross-entropy loss or log loss. There are different types of solvers that support multinomial such as 'lbfgs', 'sag' and 'newton-cg'. We are using 'lbfgs' (Limited-memory Broyden–Fletcher–Goldfarb–Shanno) solvers as it can handle multinomial loss and supports L2 regularization. For larger datasets, 'lbfgs' is really very efficient. General theorem for logistic regression is as follows -

$$logit(p) = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \ldots + b_kX_k \ldots\ldots\ldots\ldots\ldots\ldots\ldots (i)$$

After implementing logit regression, the accuracy came out around 56%.

4.3 Decision Tree Classifier

Decision Tree Classifier is a non-parametric regulated learning strategy utilized for regression & classification. It applies a direct plan of solving the classification issue. Decision Tree Classifier represents a progression of precisely made inquiries concerning the qualities of the test record. Each time it gets an answer, a subsequent inquiry is gotten some information about the class name of the record is come to. The objective is to make a model that predicts the estimation of an objective variable by taking in straightforward choice standards deduced from the data features. Basically it learns from data and then approximates a sine curve. Filtering the model will be more difficult if the decision tree is deeper. Mathematically, it can be calculated with the help of probability of the items as:

$$H = -\sum p(x) \log p(x) \ldots\ldots\ldots\ldots\ldots\ldots (\boldsymbol{ii})$$

However, accuracy of this algorithm came out about 47-48%.

4.4 Random Forest Classifier

Random forest algorithm is a supervised classification algorithm. Just like the name forest, this algorithm creates a forest with a number of trees. In machine learning, it is the most popular classification algorithm. Random forest algorithm can be used both for classification and regression kind of problems. Combining many decision trees, a random classifier can be created. The higher the number of trees in the forest gives the highest accuracy results.

4.5 K-Neighbors Classifier

Also called ask nearest neighbors, is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition from the very beginning as a non-parametric technique. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor. Here we have taken

default number of neighbors (which is 5) and used 'auto' algorithm which will attempt to decide the most appropriate algorithm based on the values passed to fit method. However, the accuracy we have got after implementing this algorithm is 58% which is the best value we have got among these seven algorithms.

4.6 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a linear transformation that is most commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications. It consists of statistical properties of collected data, calculated for each class. For a single input variable (x) this are the mean and the variance of the variable for each class. For multiple variables, this is the same properties calculated over the multivariate Gaussian, namely the means and the covariance matrix. These statistical properties are estimated from data and plug into the LDA equation to make predictions. These are the model values that we have saved to file for our model. Logistic regression is intended for two-class or binary classification problems. It can be extended for multi-class classification but is used rarely.

4.7 Gaussian NB (Naïve Bayes)

Gaussian NB classifier or Naïve Bayes classifier calculates the probabilities for every factor and after that it selects the outcome with the highest probability. This classifier also predicts the features are independent for which it is called naïve. It is considered as a powerful algorithm which is used for real time prediction, classifying text, recommendation system etc. The Bayes theorem is as follows –

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \qquad \ldots\ldots\ldots\ldots\ldots\ldots (iii)$$

We have used this classifier as it selects the outcome with highest probability. So, after implementing the classifier, we got approximate accuracy of 43%.

4.8 SVM Classifier

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyper plane. In other words, given labeled training data (supervisedlearning), the algorithm outputs an optimal hyper plane which categorizes new examples. In two dimensional

space this hyper plane is a line dividing a plane in two parts where in each class lay in either side. It uses a subset of training points in the decision function. Though it is effective in high dimensional spaces, we got an accuracy of 23.8% which is the lowest in all algorithms.

4.9 Feature Extraction

In machine learning, pattern recognition and in image processing, feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is related to dimensionality reduction. In this part, we described how we used the two types of feature extraction methods.

4.9.1 PCA Feature Reductions

Principal Component Analysis (PCA) is used to extract or reduce features and speeding up the fitting of a machine learning algorithm or to data visualization. It is used to reduce a large set of variables to a small set that will still contain most of the necessary information of the large set. To project the data in a lower dimensional space, PCA uses Singular Value Decomposition (SVD) of data for linear dimensionality reduction. It projects its components to be used on new data by implemented as a transformer object that learns ncomponents in its fitmethod. In our research we run several algorithms through our data, but the accuracy was not so satisfying. So, we decided to pick a feature reduction technique which is PCA to improve the accuracy. In PCA, we used StandardScaler method that standardized features by removing the mean and scaling to unit variance because standardization of datasets is a common requirement for most of the machine learning algorithm otherwise they may behave badly [15]. By StandardScaler, centering and scaling happen autonomously on each feature by computing the applicable statistics on the sample in the training set.

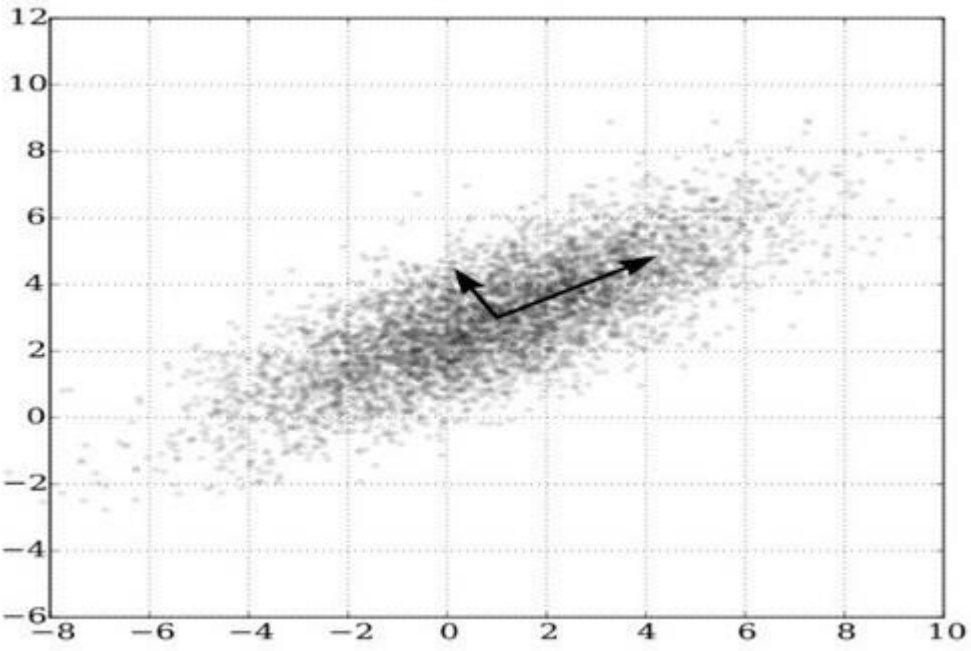

Figure 4.9.1:PCA of a multivariate Gaussian distribution centered at (1,3).

In the above figure it is shown that how PCA works. Here, the data is distributed in a two-dimensional region where the dimensions are denoted for different features. PCA finds the direction of maximal variance as well as finds the directions that are mutually orthogonal. The vectors shown are the eigenvectors of the covariance matrix scaled by the square root of the corresponding eigenvalue, and shifted so their tails are at the mean.

Parameters:

In our code the parameter that are used in PCA goes like PCA(copy=True, iterated_power='auto', n_components=0.95, random_state=None, svd_solver='auto', tol=0.0, whiten=False). The parameters for PCA stand for:

- copy: Here copyis true by default. If False, data that are passed to fit are overwritten and running fit(X).transform (X) will not yield the expected results. In that case it will use fit_transform(X). [15]
- iterated_power: The parameteriterated_poweris denoted for thenumber of iterations for the power method computed by svd_solver == ‘randomized’. [15]
- n_components: The parametern_componentsis denoted for the number of components to keep. It is basically being used for tuning. As in our code n_components=0.95, it

makes an instance of the model by choosing the minimum number of principal components such that 95% of the variance is retained. If the value (.95) is changed, the model will be changed. [15]

- random_state: The parameterrandom_state=Nonemeans the RandomState instance is the generator of random number. [15]
- Svd_solver: The parameter svd_solver='auto' means the solver is selected by a default policy based on X.shape and n_components. [15]
- tol: The parameter tolis denoted for the tolerance for singular values computed by svd_solver. [15]
- whiten: The parameter whiten is false by default. It sometimes can be used to improve the predictive accuracy of the downstream. [15]

Implementation of Machine Learning Algorithms after demonstratingPCA:

While running Logistic Regression we take two arrays, one for the raw training datasets and the other for the test datasets. After this, the train datasets are transformed from the previous test datasets using scaler.transform [15] method by retrieving the necessary features and put it in the new train array which now holds the new transformed training datasets. Same goes with the test array as now this array also holds the transformed test datasets after being transformed through scaler.transform otherwise it can't read the datasets as training is being with the modified datasets. The svm_y [15] denotes the label of the datasets. The pca.fit [15] is used to create the new model. It only can be implemented in training datasets. After all the modifications while the data become ready it was pass through Logistic Regression Algorithm. We used lbfgs solver here as the default solver was very slow. After all the modifications we found the accuracy from Logistic Regression modified through PCA is 51% which is so much improved than the previous accuracy of Logistic Regression. After running SVM with the transformed datasets by PCA the accuracy becomes 67% where previously it was 23%. RandomForestClassifierwith transformed datasets by PCA shows the accuracy of 56%. After the demonstration of K-NeighborsClassifier with transformed datasets by PCA the accuracy becomes 61% whereas previously it was 58%. After running LinearDiscriminationAnalysiswith transformed datasets by PCA the accuracy becomes 53% whereas previously it was 49%. GaussianNB Implementation shows 45% accuracy with transformed datasets by PCA whereas previously it was 42%. After running

DecisionTreeClassifier with transformed datasets by PCA the accuracy becomes 44% whereas previously it was 47%.

4.9.2 LBP Visual Descriptor

Local Binary Patterns (LBP) is a texture descriptor or visual descriptor which is usually used for classification in computer vision by feature extraction. Images are converted to grayscale as the first step in constructing the LBP visual descriptor [17]. Later, histogram is being computed over the output LBP array.

LBP creates a 3 by 3 block and looks at 9 pixels at a time. It compares every neighboring pixel with the central pixel. If the value of the any pixel is greater than 8 than it will denoted by 1, otherwise it will be denoted by 0. Binary Pattern will remain the same irrespective of illumination variation as relative difference between the pixels will remain same. Every block of that 3 by 3 block system holds so many smaller pixels and by applying histogram information are retrieved from those blocks. The transitions from 0 to 1 are the edges which show the outline of any mouth or eyelid. But, as facial expression means action, movement or differences in faces, it should be measured how these pixels changes overtime. One way is to extend the block to become a cube as by this it will have 2^{26} different possible values comparing between the center of that cube as the center has total number of 26 neighbors. To make the process more efficient, 3 orthogonal planes can be considered by which the possible solutions will be 3.2^8 which is way less than 2^{26} but the accuracy will not be hampered by this rather significant performance will be increased.

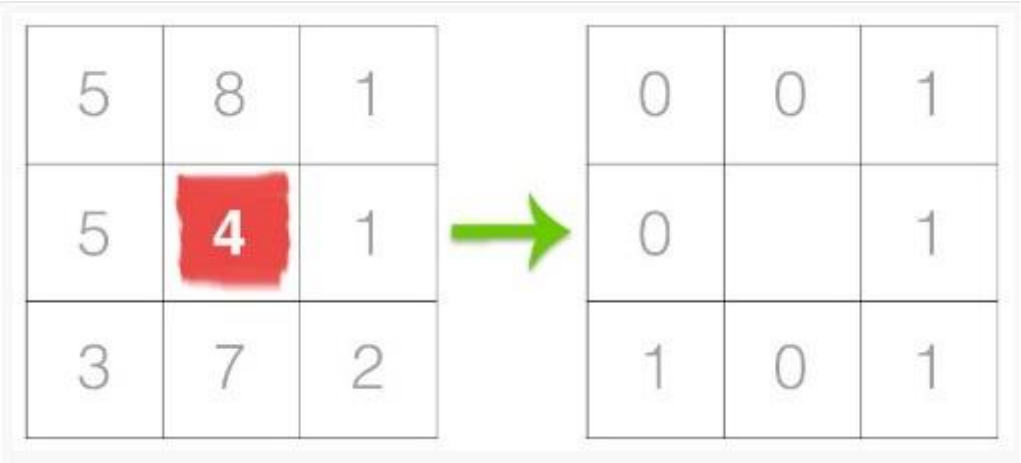

Figure 4.9.2.1: Construction of the set of 8 binary digits after the threshold of the center pixel surrounded by the 8-neighbor pixel [17].

In the above figure, the center pixel is compared with its neighboring 8 pixels. If the intensity of the center is greater than its neighbor than we denote it with the binary digit 1, or else 0. As there are 8 surrounded pixels, there will be a 2^8-possible solution of LBP [17].

Implementation of LBP

We used the scikit-image of LBPs as they offer more control of the types of LBP histogram[17]. Moreover, the scikit-image implementation additionally incorporates variants of LBPs that improves rotation and grayscale invariance. Our goal was to extract Local Binary Patterns from our datasets and apply some machine learning algorithm to automatically recognize and categorize our data.At start we imported the feature sub-module of scikit-image which contains the implementation of the Local Binary Patterns descriptor.Also, we imported LinearSVC.

After that we gathered our data into and array to train our classifier. Then we labeled the data. The label of the array is the label of every image which is used to store the name of each images. We used the previous 4-dimensional array of our data here. Then we need to convert them in grayscale. In the line img = cv2.imread(path, cv2.IMREAD_GRAYSCALE) the images are loaded from the path and then converted into grayscale. After this we resized the data. The lbp= feature.local_binary_pattern(img, 24, 8, method='uniform") is for grayscale and rotation invariant LBP. The parameters are image which is for graylevel image, 24 is denoted for the number of circularly symmetric neighbor set points which is need for quantization of the angular space, 8 denotes the radius of the circle which is actually the spatial resolution of the operator and lastly method ="uniform" which means improved rotation invariant with uniform patterns[18].

Later, we normalized our histogram data. In the field of image processing normalization is done to change the range of pixel intensity values. It is used to bring the image or other type of signals into a range that is more familiar or normal to the senses. At the end prediction=model.predict(hist_test) is used to make predictions on new data instances after choose and fit the final machine learning model in scikit-learn[18].

After the demonstration of LBP, we again run all the machine learning algorithm and found that the accuracy is being increased than the accuracy before any feature extraction was constructed.But again, we found most accurate result by demonstrate PCA, so we are considering that.

CHAPTER 5

Experiment and Performance Analysis

We have implemented seven algorithms separately which includes- Random Forest Classifier, Support Vector Machine, Logistic Regression, Decision Tree Classifier, K-Neighbors Classifier, Linear Discriminant Analysis and Gaussian Naïve Bayes. First, we labeled all the collected data, trained and tested them and then ran all these algorithms and classifiers. We have also used PCA with all other raw algorithms and LBP with other raw algorithms.

5.1 CNN(Convolutional Neural Network)

We made a model in CNN with our own dataset. After testing the Test_Data we got around 56% accuracy. Below we haveshown the internal loss and the accuracy graph.

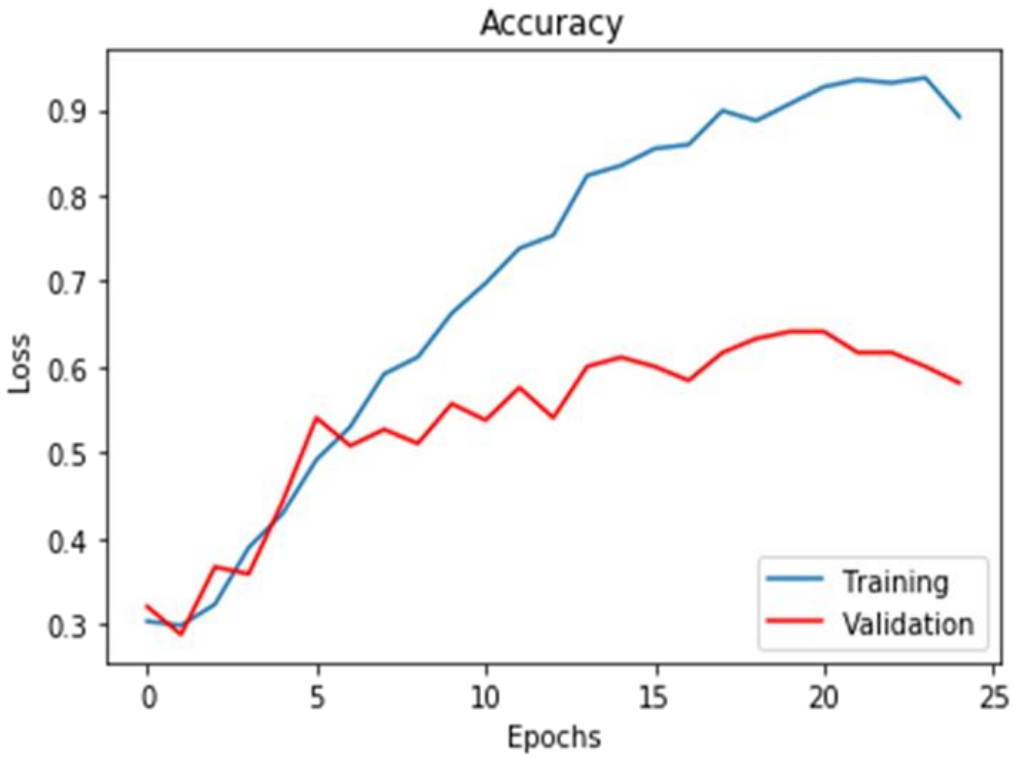

Fig 5.1.1: The internal Accuracy in every Epochs.

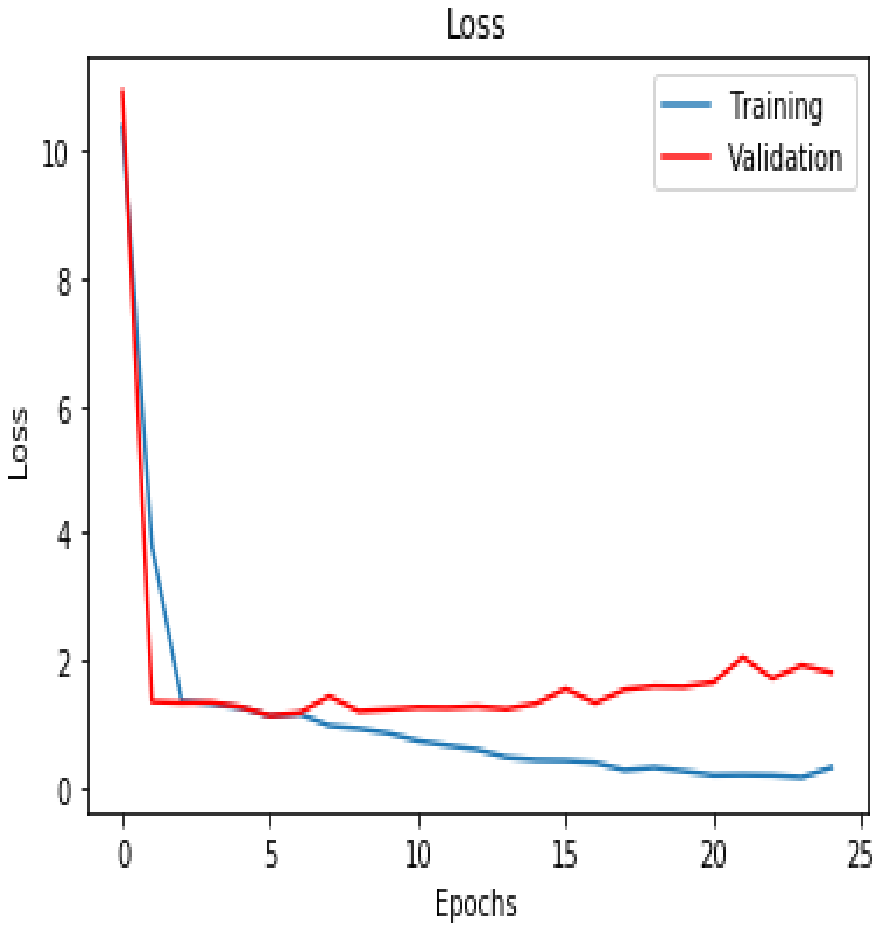

Fig 5.1.2: The internal Loss in every Epochs

We run 25 iteration. the internal values of the iteration is showed in.

5.2 Algorithms without Feature Extraction

Here, we have run all seven algorithms and classifiers separately. We can see among these seven algorithms, the highest accuracy we got from K-Neighbors classifier (which is 61.5%) whereas the lowest value is of SVM (Support Vector Machine) Classifier which is 25.4%.

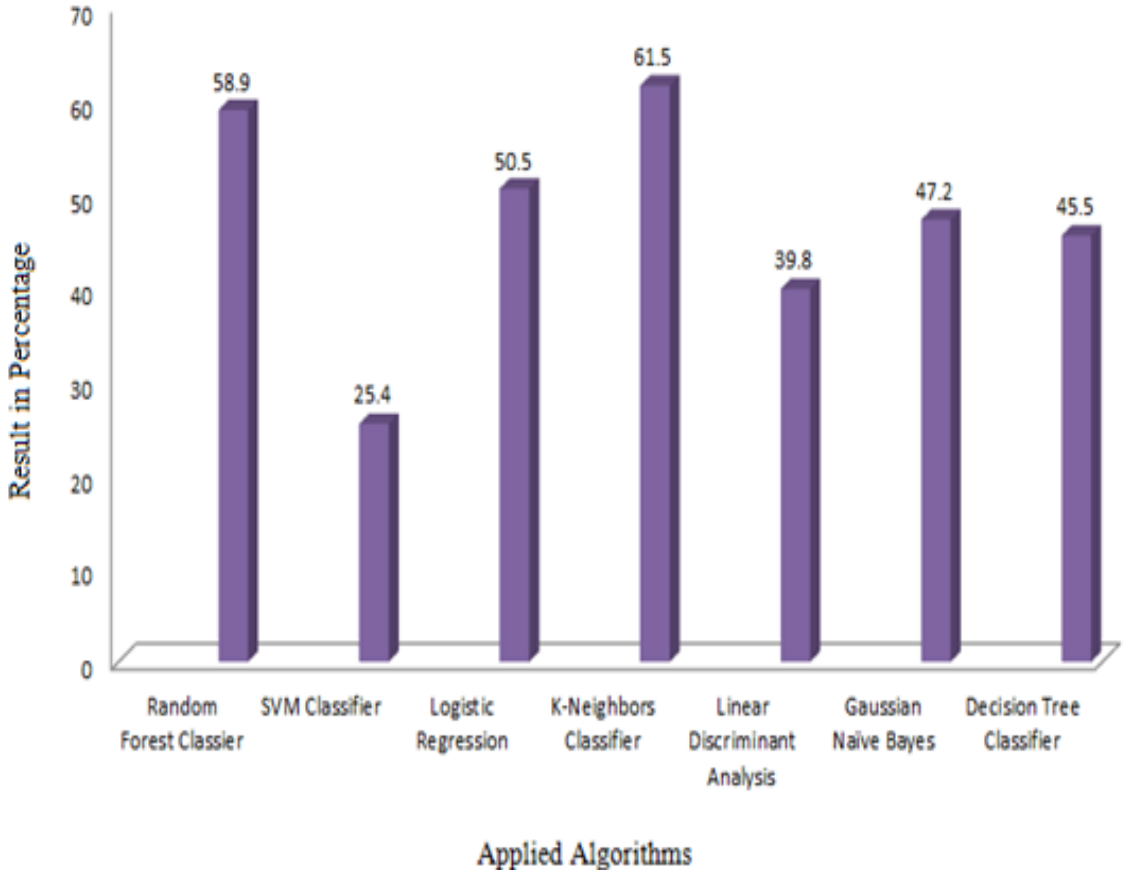

Fig 5.2: All algorithms without feature extraction.

5.2.1 Implementing Random Forest Classifier along with PCA & LBP

The figure below shows the accuracy comparison of Random Forest Classifier, PCA with Random Forest and LBP with Random Forest. Highest accuracy is 58.9% and the lowest is 48.2%.

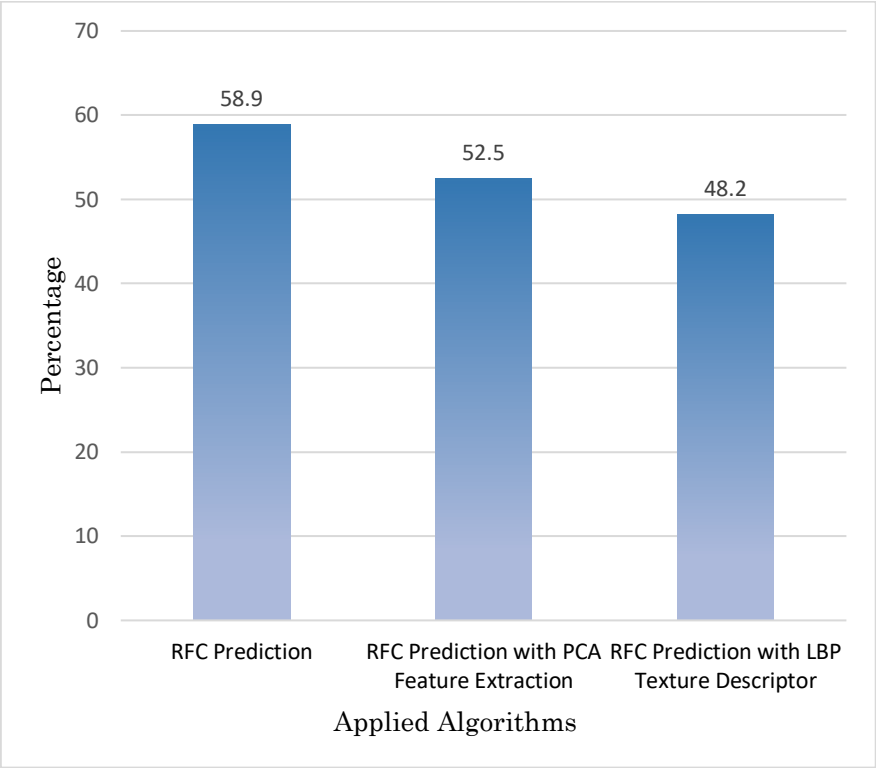

Fig 5.2.1: Random forest classifier with PCA & LBP

5.2.2 Implementing SVM Classifier along with PCA & LBP

SVM classifier, more precisely Support Vector Machine Classifier gives us the lowest accuracy of all other implemented classifiers which is 25.4% whereas SVM along with PCA gives us the highest accuracy that is 68.2%.

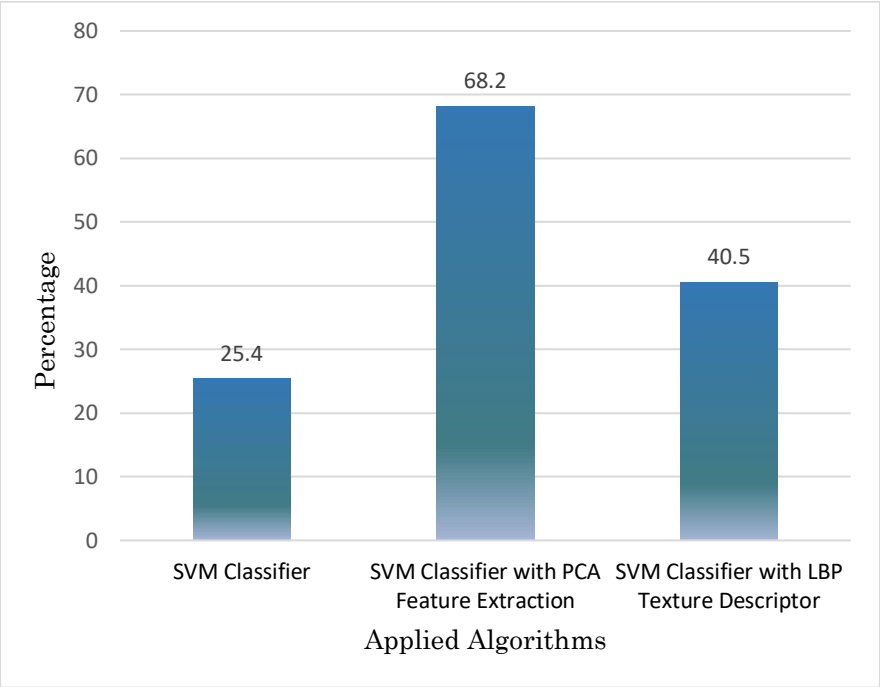

Fig 5.2.2: SVM classifier with PCA & LBP.

5.2.3 Implementing Logistic Regression along with PCA

In the figure below, Logistic Regression alone gives accuracy of 50.5% and along with PCA, it has an accuracy of 57.9%.

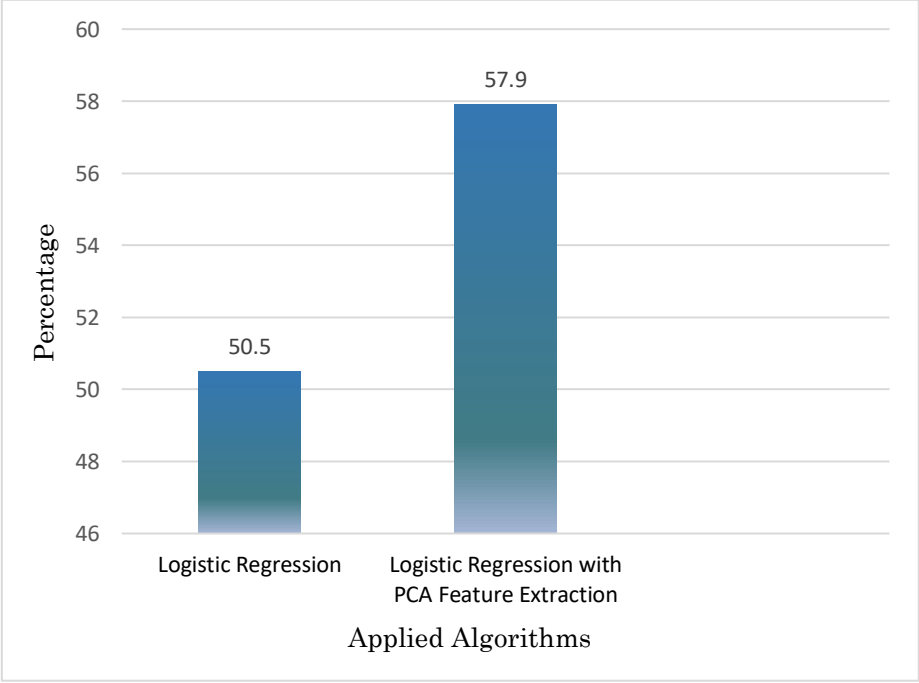

Fig 5.2.3: Logistic Regression with PCA feature extraction.

5.2.4 Implementing K-Neighbors Classifier along with PCA & LBP

K-Neighbors Classifier has the accuracy of 61.9% and with PCA the accuracy goes a little up that is 58.9%. But the lowest accuracy we got is 49.2%, from LBP along with K-Neighbors Classifier.

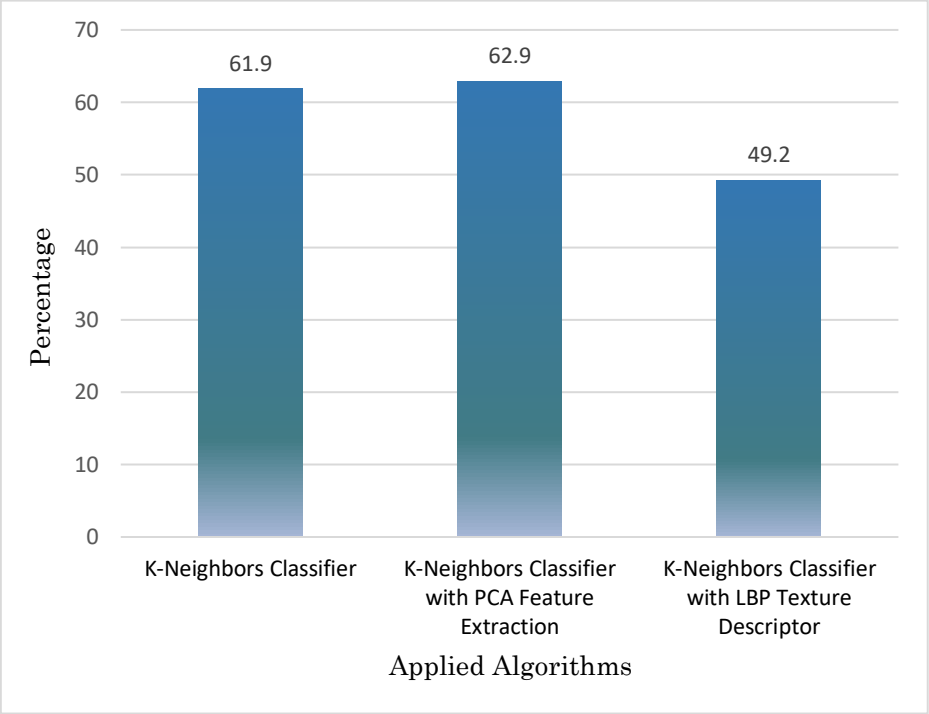

Fig 5.2.4: K-Neighbors Classifier with PCA & LBP.

5.2.5 Implementing Linear Discriminant Analysis along with PCA & LBP

Linear Discriminant Analysis alone gives the accuracy of 39.8% which is lowest in terms of implementing with PCA & LBP. With PCA, we get the accuracy of 58.9% and with LBP the accuracy goes down a bit of 40.5%.

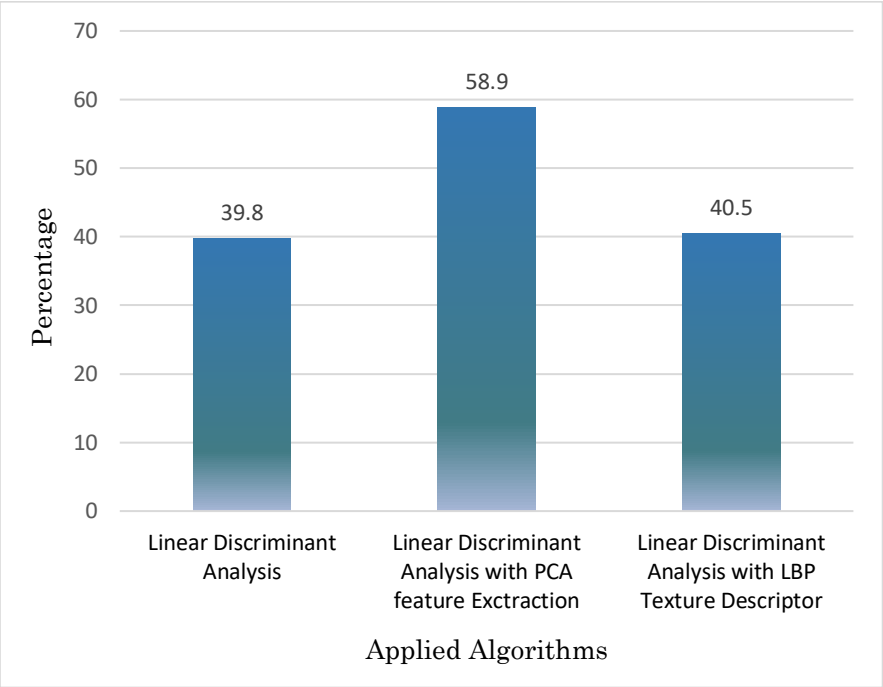

Fig 5.2.5: Linear Discriminant Analysis with PCA & LBP.

5.2.6 Implementing Gaussian NB (Naïve Bayes) along with PCA & LBP

Gaussian NB gives the accuracy of 47.2%; with PCA the accuracy changes into 40.8% and with LBP the accuracy we got is 42.8%.

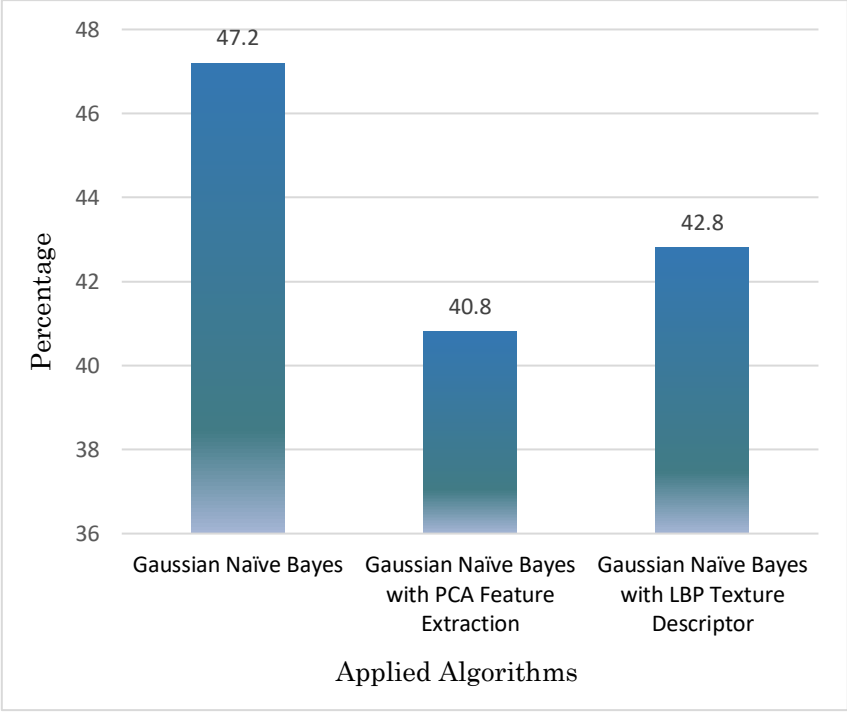

Fig 5.2.6: Gaussian Naïve Bayes with PCA & LBP.

5.2.7 Implementing Decision Tree Classifier along with PCA & LBP

We have used a total number of 100 decision trees for our data and the accuracy we got from DTC alone is 45.5%, 48.8% with PCA and 35.5% with LBP.

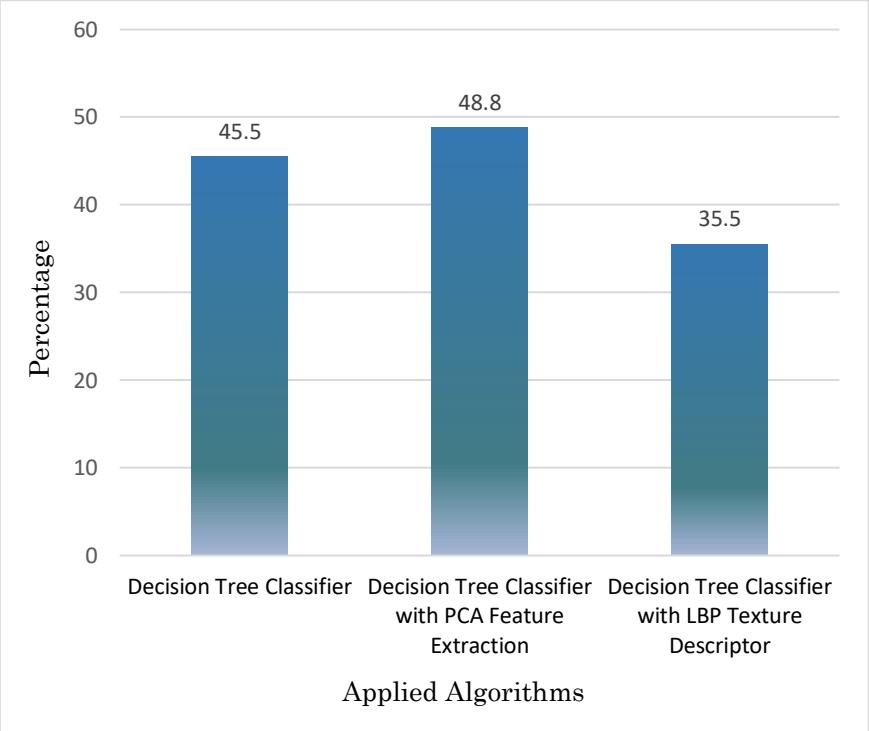

Fig 5.2.7: Decision Tree Classifier with PCA & LBP.

5.3 PCA(Principal Component Analysis) with all seven algorithms

In this phase, we have implemented all other algorithms and classifiers along with PCA and the highest accuracy is from PCA with SVC which is 68.2% and the lowest is 40.8%, from PCA along with Gaussian NB.

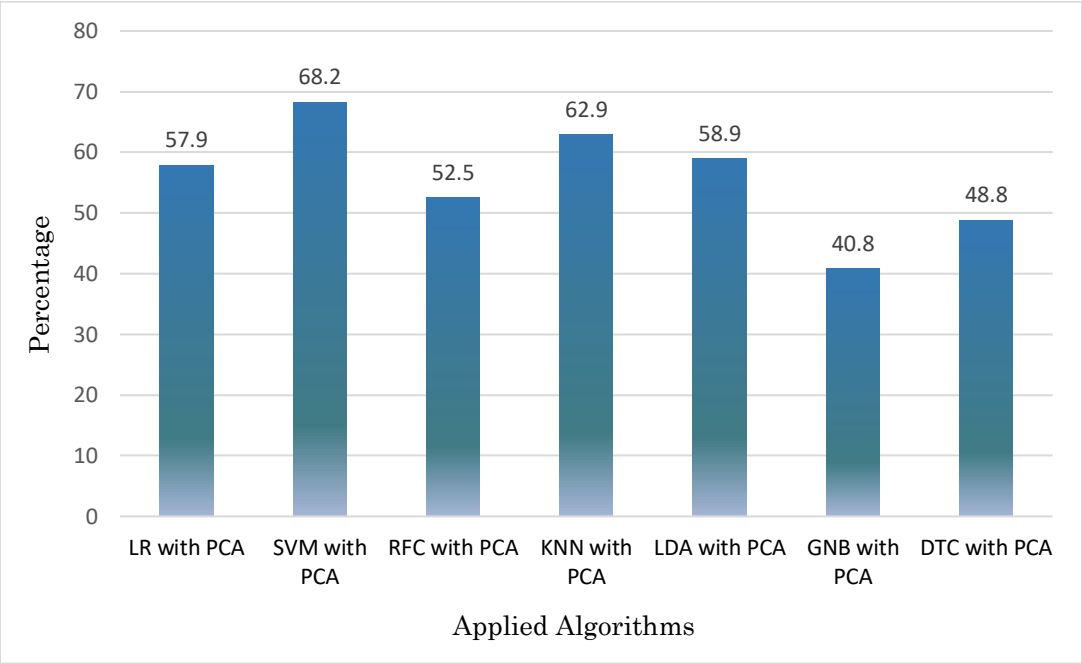

Fig 5.3: PCA Feature Extraction along with all seven algorithms.

5.4 LBP (Local Binary Pattern) with all algorithms

The highest accuracy is 49.2%, from LBP with KNN and the lowest accuracy is from LBP with Decision Tree Classifier which is 35.5%.

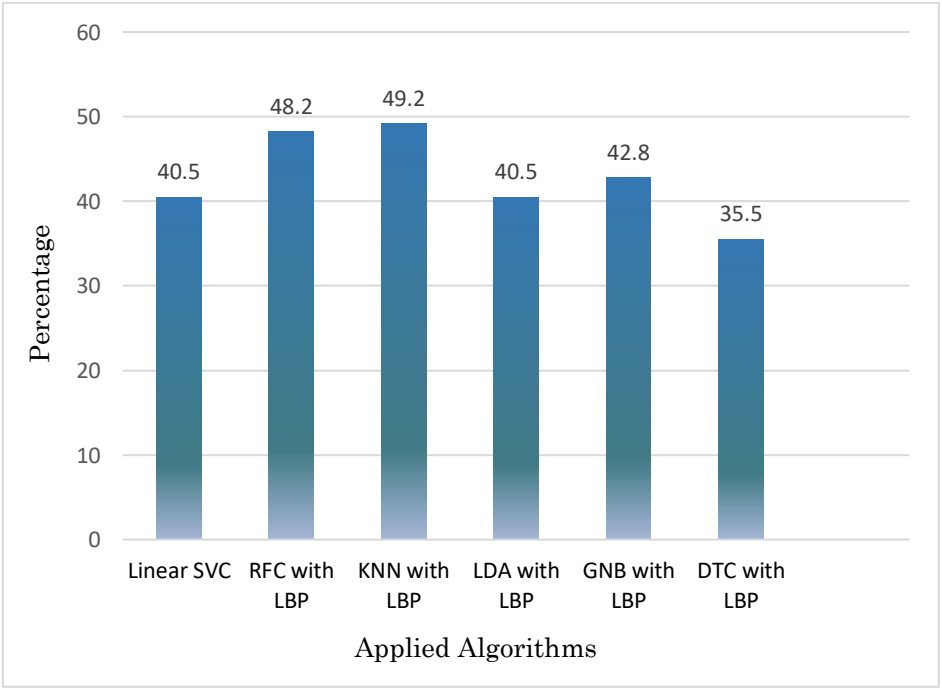

Fig 5.4: All algorithms along with LBP.

5.5 Overall Performance Analysis Comparison

Therefore, after comparing all algorithms with PCA and LBP, alone SVM Classifier gives us the lowest accuracy of 25.4% whereas the highest accuracy we are getting is after implementing SVC with PCA that is 68.2%.

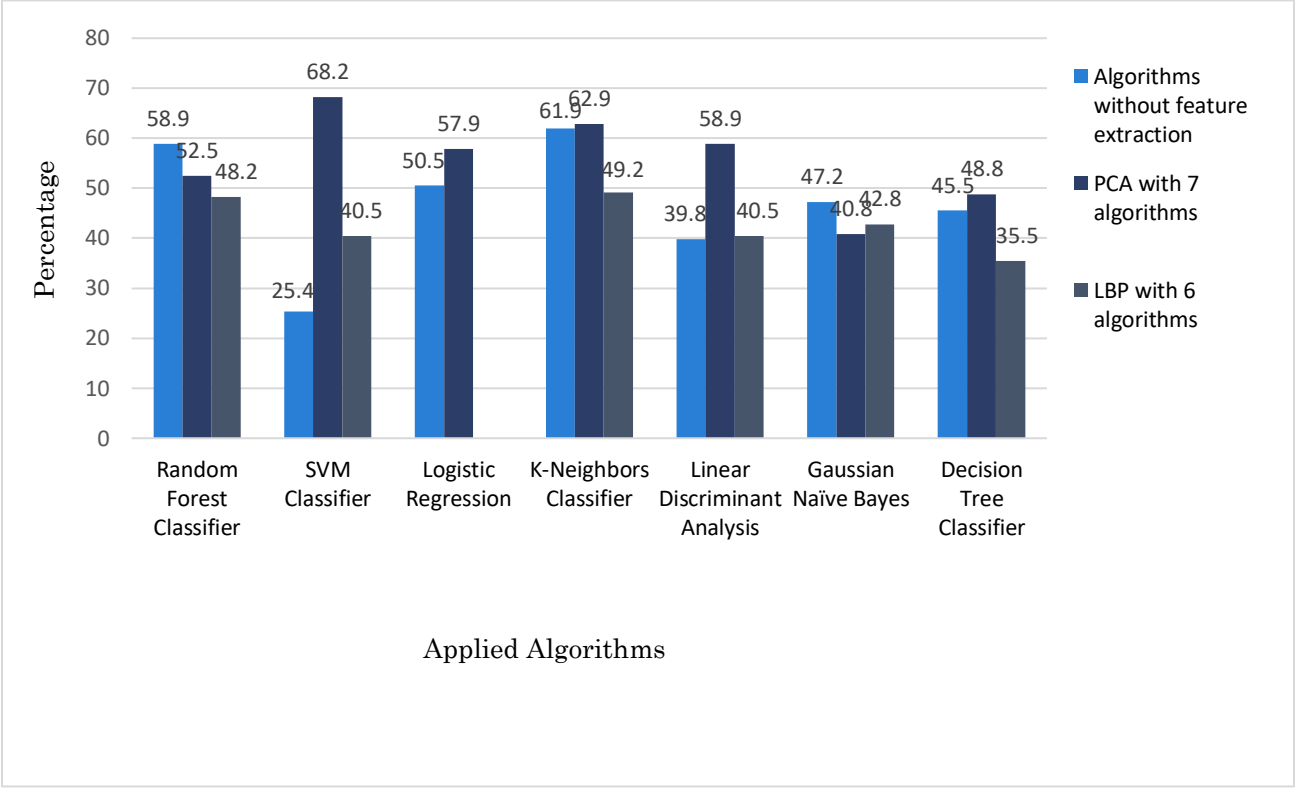

Fig 5.5: Comparisons of all algorithms along with PCA & LBP.

After final compilation of all the algorithms and classifiers, the result came out as follows-

Feature Extraction Techniques	**RFC**	**SVM**	**LR**	**KNN**	**LDA**	**GNB**	**DTC**
Algorithms without feature extraction	58.9	25.4	50.5	61.9	39.8	47.2	45.5
PCA	52.5	68.2	57.9	62.9	58.9	40.8	48.8
LBP	48.2	40.5	___	49.2	40.5	42.8	35.5

Table 5.1: Computed final scores in percentage

CHAPTER 6

Conclusions and Future Research

Results& Discussion

The first experimental phase was aimed at analyzing all the children's images, training those datasets and testing them. The overall facial complexity was computed by averaging all the obtained entropy values. The entropy values computed on the overall actionunits are very interesting.These results are broadlyconsistent with those expected considering bothclinical studies or experimental evidences derived throughinvasive methods for data acquisition. This way theoutcomes of the proposed non-invasive approach pone a interesting perspective to make possible the analysis of facialdynamics by using just machine learning based algorithms.In the second experimental phase we applied all the seven algorithms separately and then applied them again with the feature extraction methods. The result you can see in Table 5.1 includes all the accuracies we have got so far. The outcome means that, through deeper experimental investigations, the introduced approach could bring to affordable non-invasive measurement for the diagnosis and assessment of autism spectrum disorders.

6.1Conclusion

Being the pioneer in this particular research field, the whole journey was both inspiring and challenging for us. Till now ASD is a mysterious pattern of unusual behavior which has no precautions or remedy. Scientists are still struggling with the cause and effect of this special behavioral disorder. Every ASD child has individual specialty as researchers and doctors claim. Our target was to find out if there is any pattern of their behavior by which their emotions can be understood by others. For fulfilling this desire, we choose several well renowned machine learning algorithms as well as the popular image processing algorithm CNN and try to find out if these algorithms can keep their promises to perform for ASD people’s emotion recognition as efficiently as they perform for ordinary people. The procedure we followed to find it out was comparing between several algorithms on the basis of which algorithm gives the highest accuracy and also which emotions of ASD children are well recognized. We actually followed the same procedure that was followed in so many reputed prior researches to retrieve emotions from facial expressions. The only difference was that our datasets was of ASD children’s. We believe at that point we have added a new dimension in the image processing and emotion recognition research field. We have already done some mentionable progress to achieve our ultimate goal. We strongly

believe further research in this field based on our work can find a pattern for their behavior. By the pattern a list of categories can be made of their facial expression for different emotions that will make it easier for everyone to deal with them and by this the ultimate goal will be achieved.

6.2Future Work Plan

This work of ours introduced a proposal of machine-learning strategies during computer-ASD children interactions in order to make possible an objective evaluation of children's behaviors and then to give the possibility to introduce a metric about the effectiveness of the therapy. In particular, the work focused on the basic emotion recognition skills and it contributed to introduce facial expression recognition (FER) engine that will automatically detect and track the child's face and then recognize emotions on the basis of a machine learning pipeline based on Inception. Life planning is essential for persons with developmental disabilities. They will need help to plan for their lifetime care, supervision, protection and financial security. It is a tragedy when children and adults with autism are not able to fully participate in their communities because they cannot access the services that would allow them to do so. The more we learn about autism, the more hope we have for treatment. We have the power to change for the better any given situation by applying what we are learning. Our future work will also deal with evaluating the systems along with multiple therapeutic sessions involving the same children in order to take advantage of the analysis tools implemented by the meta-data handling module.

References:

[1] TensorFlow. (2018). Build a Convolutional Neural Network using Estimators | TensorFlow. [online] Available at: https://www.tensorflow.org/tutorials/estimators/cnn [Accessed 15 Jul. 2018].

[2] Williams, Travis & Li, Robert. (2016). Advanced Image Classification Using Wavelets and Convolutional Neural Networks. 10.1109/ICMLA.2016.0046.

[3] Dachapally, P. (2016). Facial Emotion Detection Using Convolutional Neural Networks and Representational Autoencoder Units. [online] Arxiv.org. Available at: https://arxiv.org/pdf/1706.01509.pdf [Accessed 16 Jul. 2018].

[4] S. Alizadeh and A. Fazel, "Convolutional Neural Networks for Facial Expression Recognition",Cs231n.stanford.edu,2017.[Online].Available: http://cs231n.stanford.edu/reports/2016/pdfs/005_Report.pdf. [Accessed: 16- Jul- 2018].

[5] B. Zhang, C. Quan and F. Ren, "Study on CNN in the recognition of emotion in audio and images - IEEE Conference Publication", Ieeexplore.ieee.org, 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7550778/. [Accessed: 16- Jul- 2018].

[6] E. Bal, E. Harden, D. Lamb, A. Van Hecke, J. Denver and S. Porges, "Emotion Recognition in Children with Autism Spectrum Disorders: Relations to Eye Gaze and Autonomic State", *Journal of Autism and Developmental Disorders*, vol. 40, no. 3, pp. 358-370, 2010.

[7] S. Kuusikko, H. Haapsamo, E. Jansson-Verkasalo, T. Hurtig, M. Mattila, H. Ebeling, K. Jussila, S. Bölte and I. Moilanen, "Emotion Recognition in Children and Adolescents with Autism Spectrum Disorders", *Journal of Autism and Developmental Disorders*, vol. 39, no. 6, pp. 938-945, 2009.

[8] "About Train, Validation and Test Sets in Machine Learning", *Towards Data Science*, 2018. [Online]. Available: https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7. [Accessed: 18- Jul- 2018].

[9] "What is the meaning of flattening step in a convolutional neural network? - Quora", *Quora.com*, 2018. [Online]. Available: https://www.quora.com/What-is-the-meaning-of-flattening-step-in-a-convolutional-neural-network. [Accessed: 18- Jul- 2018].

[10] M. Uljarevic and A. Hamilton, "Recognition of Emotions in Autism: A Formal Meta-Analysis", *Journal of Autism and Developmental Disorders*, vol. 43, no. 7, pp. 1517-1526, 2012.

[11] O. Golan, E. Ashwin, Y. Granader, S. McClintock, K. Day, V. Leggett and S. Baron-Cohen, "Enhancing Emotion Recognition in Children with Autism Spectrum Conditions: An Intervention Using Animated Vehicles with Real Emotional Faces", *Journal of Autism and Developmental Disorders*, vol. 40, no. 3, pp. 269-279, 2009.

[12] O. Golan, S. Baron-Cohen, J. Hill and M. Rutherford, "The 'Reading the Mind in the Voice' Test-Revised: A Study of Complex Emotion Recognition in Adults with and Without Autism Spectrum Conditions", *Journal of Autism and Developmental Disorders*, vol. 37, no. 6, pp. 1096-1106, 2006.

[13] N. Bauminger, *Journal of Autism and Developmental Disorders*, vol. 32, no. 4, pp. 283-298, 2002.

[14] Wahed, R. B. and Nivrito, AKM (2016), Comparative Analysis between Inception-v3 and Other Learning Systems using Facial Expressions Detection.

[15] http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

[16] https://en.wikipedia.org/wiki/Principal_component_analysis.

[17] https://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/

[18]http://scikitimage.org/docs/dev/auto_examples/features_detection/plot_local_binary_pattern.htmlhttps://en.wikipedia.org/wiki/Local_binary_patterns.

[19] Adolphs, R., Sears, L., &Piven, J. (2001). Abnormal processing of social information from faces in autism, Journal of Cognitive NS.

[20] M. Uddin, J. Lee, and T.-S. Kim. An enhanced independent component-based human facial expression recognition from video. Consumer Electronics, IEEE Transactions on, 55(4):2216–2224, November 2009.

[21] Corden, B., Chilvers, R., &Skuse, D.(2008). Avoidance of emotionally arousing stimuli predicts social-perceptual impairment in AS.

[22] M. Siddiqi, R. Ali, A. Khan, E. Kim, G. Kim, and S. Lee.Facial expression recognition using active contour-based face detection, facial movement-based feature extraction, and non-linear feature selection.Multimedia Systems, pages 1–15, 2014.

[23] A. Ramirez Rivera, R. Castillo, and O. Chae. Local directional number pattern for face analysis: Face and expression recognition. Image Processing, IEEE Transactions on, 22(5):1740–1752, May 2013.

[24] A. Klin, D. J. Lin, P. Gorrindo, G. Ramsay, and W. Jones. Two-year-olds with autism orient to non-social contingencies rather than biological motion. Nature, 459(7244):257– 261, 05 2009.

[25] W. Zhen and Y. Zilu. Facial expression recognition based on local phase quantization and sparse representation. In Natural Computation (ICNC), Eighth International Conference on, pages 222–225, May 2012.

[26] A. Peca, R. Simut, S. Pintea, C. Costescu, and B. Vanderborght. How do typically developing children and children with autism perceive different social robots? Computers in Human Behavior, 41:268–277, 2014.

[26] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society Conference on, volume 1, pages 886–893 vol. 1, June 2005.

[27] S. M. Anzalone, E. Tilmont, S. Boucenna, J. Xavier, A.-L.Jouen, N. Bodeau, K. Maharatna, M. Chetouani, and D. Cohen. How children with autism spectrum disorder behave and explore the 4-dimensional (spatial 3d + time) environment during a joint attention induction task with a robot. Research in Autism Spectrum Disorders, 8(7):814 – 826, 2014.

[28] A. Dhumane, R. Prasad, J. Prasad. 2016. *Routing Issues in Internet of Things: A Survey*. In Proceedings of the International MultiConference of Engineers and Computer Scientists 2016, Hong Kong, vol. 1.