ASHESI

ASHESI UNIVERSITY

Low Cost, Portable Facial Emotion Recognition Device for Autistic Children

CAPSTONE PROJECT

BSc. Electrical/Electronic Engineering

JEAN EWURAMA ROBERTS

2021

ASHESI UNIVERSITY

Low Cost, Portable Facial Emotion Recognition Device for Autistic Children

CAPSTONE PROJECT

Capstone Project submitted to the Department of Engineering, Ashesi University in partial fulfillment of the requirements for the award of Bachelor of Science degree in Electrical/Electronic Engineering.

Jean Ewurama Roberts

2021

DECLARATION

I hereby declare that this capstone is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature: ...............................J.R..

Candidate's Name: Jean Ewurama Roberts

Date: April 27, 2021

I hereby declare that preparation and presentation of this capstone were supervised in accordance with the guidelines on supervision of capstone laid down by Ashesi University.

Supervisor's Signature: ....

Supervisor's Name: ....

Date: ....

ACKNOWLEDGMENTS

I would like to express my gratitude to God for his faithfulness and goodness over the entire duration of this project.

I would also like to thank my family, friends and my supervisor, Dr. Siby Samuel for their guidance and direction.

ABSTRACT

This paper details the use of a Raspberry Pi 4 to develop a facial emotion recognition device for autistic children. This device will be used to recognize seven basic emotions: Happiness, Anger, Sadness, Disgust, Fear, Surprise and Neutrality. The face detection algorithm used is the Viola-Jones face algorithm whilst the feature extraction algorithm used is a fusion of local binary patterns and histogram of oriented gradients. The machine learning algorithm used for classification is the Linear Support Vector Machine. This system produced an accuracy of 35.68% with the most accurately classified emotion being happy. This paper investigates the use of both local binary patterns and histogram of oriented gradients as feature extraction methods as well as both Support Vector Machine and Multilayer perceptron neural network as classification methods.

TABLE OF CONTENTS

INTRODUCTION

According to The Center for Disease Control and Prevention, Autism spectrum disorder (ASD) is a developmental disability that can cause significant social, communication and behavioral challenges (n.d.). Unfortunately, there is not enough information about autism in Ghana. "Students with autism are included in the categories, "Intellectual Disability (I.D.)" or "Mentally handicapped" for educational purposes". (Ofori-Atta & Ohene, 2014). However, in the United States of America, The Centre for Disease Control estimates that autism affects every 1 in 54 children.

"Children with Autism Spectrum Disorder often find it difficult to recognize facial expressions and the emotion behind it, copy or use emotional expressions, understand and control their own emotions, understand and interpret emotions – they might lack, or seem to lack empathy with others." ("Emotional development in children with autism spectrum disorder", 2020). In 1943, the first physician to conduct research in Autism, Leo Kanner "viewed their inability to form the usual, biologically provided affective contact with people as a primary feature of the disorder" (Kanner,1943). Though some of his research has been modified, this has remained a major feature of the Autistic Spectrum Disorder.

The emotion recognition impairment was further confirmed in Kieran M. Rump, Joyce L. Giovanneli, Nancy J. Minshew and Mark S. Strauss, "The development of emotion recognition in individuals with Autism". Both individuals on the autism spectrum and those not on the spectrum participated in an emotion recognition experiment. The medium for testing was brief video displays of facial expressions. For the first age group of children between 5-7 years, it was observed that autistic children performed worse than normal-functioning children. For the two remaining age groups, 8-12 years, and 13-17 years, it was observed that both age groups had similar results, though still worse than normal-functioning children.

This poses the next question, "can emotion recognition be taught to children with autism spectrum conditions", and this was answered by Simon Baron-Cohen, Ofer Golan and Emma Ashwin using a case study of The Transporters, an animated series designed to enhance emotion comprehension in children on the Autism Spectrum. Autistic Children between the ages of 4-7 years old watched the Transporters every day for four weeks. At the end of the period, they participated in an emotion recognition task together with autistic children that had not gone through the experiment. It was observed that the children that watched the transporters performed better than those that didn't. This informs us that emotion recognition can be taught to children with Autism Spectrum Disorder.

Though individuals on the spectrum have a problem with emotion recognition, they have enhanced skills in systemizing. "Systemizing is the drive to analyze or build systems, allowing one to predict the behavior of the system and control it. These include vehicles, spinning objects and computers" (Baron-Cohen et al, 2009). Using these capabilities of individuals on the Autism Spectrum, the Transporters embeds the emotion recognition learning process in a world of mechanical vehicles.

However, there is a method that utilizes a DVD known as "MindReading" proposed by Baron-Cohen et al. "It was developed to help people with Autism Spectrum Disorder learn to recognize both basic and complex emotions and mental states from video clips of facial expressions and audio recordings of vocal expressions." (Baron-Cohen et al, 2009). Individuals on the spectrum might struggle to recognize emotions in real-time due to the fast nature in which emotions change. With the mindreading DVD, the computer-based environment enables the child on the spectrum to have control and have enough time to study the unique features of each emotion. Another problem with emotion recognition in real-time in live social situations is that every individual might express their emotions differently and this might serve as an obstacle during the emotion learning process with children on the spectrum. The Mindreading DVD prevents this problem by having the emotions displayed by six different actors. Over a 10-week period, it was observed that individuals on the spectrum that watched the Mindreading DVD reported an improved ability of emotion recognition, even after a year. A future work proposed by Baron-Cohen et al. is to assess the benefit of a longer intervention period than 10 weeks.

In this paper, a low cost, portable facial emotion recognition device is proposed that will increase the intervention period to improve the learning curve of facial emotion recognition in autistic children.

LITERATURE REVIEW

Much work has been done in the field of facial emotion recognition. Though much progress has been made, recognizing facial emotions with a high accuracy is difficult due to the complexity and variety of facial expressions. Other factors that also come into play are variation in light intensity, the camera's focus and pose variations.

There are seven basic emotions usually classified in the facial emotion recognition process: happiness, anger, surprise, fear, disgust, sadness and neutral (the absence of emotions). Images showing these seven basic emotions are gotten from datasets such as the Extended Cohn-Kanade (CK+), Japanese Female Facial Expression (JAFFE), FER2013 and Radboud faces database.

The facial emotion recognition process includes three main processes: face recognition, feature extraction and classification. However, before images go through this process, they have to be pre-processed, of which in most papers includes the face recognition step. However, in Caifeng Shan et.al, "Robust facial expression recognition using local binary patterns ", the faces were normalized to a fixed distance of 55 pixels between the centers of the two eyes. The images were cropped according to this normalization.

Viola-Jones is the most popular algorithm used for face recognition because of its high detection accuracy and real-time performance. It uses Haar cascades to recognize the face and it can be implemented on different devices as demonstrated by Suchitra et al., that implemented it on a raspberry pi II and Rohit Verma et al., Bayezid Islam et al., S.L. Happy et al. and Pranav Kumar et al. that implemented it on the central processing unit of a computer. M Murugappan et al. improved the accuracy of the Viola-Jones algorithm using an Adaboost classifier in his paper,"Facial Expression Classification using KNN and Decision Tree Classifiers".

Facial segmentation or facial landmark detection is a method employed before feature extraction to reduce the computational cost and reduce the over fitting of the features for classification. Haar cascades used by the Viola Jones algorithm can also be used for detecting facial landmarks such as the mouth and eyes as used by Priyanka Nair et al and M Murugappan et al. However, in addition to the Haar cascades used for the face and eye location, M Murugappan et al. placed facial action units on specific locations on the subject's face using the facial action coding system to improve their system since it was being operated in real-time. Bayezid Islam et al. segmented the pre-processed facial images into four segments: right eye, left eye, nose, and mouth. Although, S.L. Happy et al. did not segment the face, they extracted features from active facial patches, which are facial regions that undergo a major change during different expressions. Ten active patches were selected: Three were located around the lips, four on the cheek, two on the forehead and one in the upper nose region. These active patches were located by localizing some of the facial landmarks such as eyes, nose, and lips corners accurately. Unlike the facial segmentation and active facial patches method, Pranav Kumar et al. used an ensemble of randomized regression trees to detect 68 facial landmarks to reduce the computational cost of the system.

The next step in the facial emotion recognition process is feature extraction. There are two main feature extraction methods, appearance-based methods, and geometry-based method. For geometry-based feature extraction, facial segmentation is not necessary. One popular geometry-based feature extraction method is the Active Shape Model as used by Suchitra et al, where 26 facial features were extracted. This feature extraction method together with an Adaboost classifier had an average accuracy of 94% in real-time with five out of the seven basic emotions (anger, disgust, happiness, neutral and surprise) on a raspberry pi II. However, most papers employ appearance-based feature extraction methods as they are more efficient, faster, and more accurate with low resolution images. There are many appearance-based feature extraction methods such as the gray level difference method as used by Priyanka Nair et al, Local Binary Pattern and Histogram of Oriented Gradients. Local Binary Pattern features, in particular, are known for their excellent light invariance property and low computational complexity.

In Rohit Verma and Mohamed-Yahia Dabbagh “Fast facial expression recognition based on local binary patterns”, A recognition accuracy of 86.67% was achieved with local binary patterns as the feature extraction method and support vector machine and Adaboost classifier as the classifier.

In Shu Liao, Wei Fan, Albert C.S. Chung, Dit-Yan Yeung “Facial Expression Recognition using Advanced local binary patterns, Tsallis entropies and Global Appearance Features”; two sets of features were extracted: texture features and global appearance features. The texture features were extracted using extended local binary patterns in both intensity and gradient maps and computing the Tsallis entropy of the Gabor filtered response. The second set of features is extracted by performing null space based linear discriminant analysis on the training face images. This approach was tested on static images and the classifier used was a support vector machine with a rbf kernel. The accuracy of this method was 80.4%

In Caifeng Shan, Shaogang Gong and P.W. McOwan “Robust facial expression recognition using local binary patterns”, simple local binary patterns were used to represent salient micro-patterns of face images. A recognition result of 79.1% was obtained together with a support vector machine. In this paper, local binary patterns as a feature extraction method was compared with a geometric features based TAN classifier and Gabor Wavelets. LBP had a higher accuracy than all the other methods on the Cohn-Kanade dataset.

In S L Happy, Aurobindo Routray, “Robust facial expression classification using shape and appearance features”, both shape and appearance features were extracted to form a hybrid feature vector. Pyramid of Histogram of Gradients was used as shape descriptors whilst Local

Binary patterns were used as appearance features. Linear discriminant analysis, a dimensionality reduction process is used to reduce the dimensionality of the features and these are classified using the Support Vector Machine. This proposed method was tested on two different datasets; CK+ and JAFFE and produced a higher accuracy on the CK+ dataset

In Bayezid Islam, Firoz Mahmud, Arfat Hossain "High Performance Facial Expression Recognition System using facial region segmentation, Fusion of HOG & LBP Features and Multiclass SVM", both histogram of oriented gradients and local binary patterns features were extracted, and the dimensionality of this combined feature vector was reduced using principal component analysis. The accuracy of this feature extraction method, together with a multiclass support vector machine on JAFFE dataset was 94.42%, 99.59% on CK+ dataset and 99.65% on RaFD.

In Pranav Kumar, S L Happy, Aurobinda Routray "A real-time robust facial expression recognition system using HOG features", histogram of oriented gradients (hog) features are extracted from active facial patches to improve the robustness of the system against scale and pose variations. An accuracy of 95% was achieved with this feature extraction method and support vector machine classification method on the Cohn-Kanade Data set.

In Sushanta Roy Supta, Md. Rifath Sahriar, Md. Golam Rashed, Dipankar Das and Rabaiyat Yasmin, "An effective facial expression recognition system", an automated facial expression recognition system based on Histogram of oriented gradients and support vector machine with a polynomial kernel is proposed. This proposed method is evaluated on the JAFFE database and Cohn-Kanade facial expression database. The accuracy was revealed to be 97.62% on the JAFFE database and 98.61% on the Cohn-Kanade database.

Though extracting features is a very important stage of the facial emotion recognition process, the classification algorithm can either boost the accuracy of the feature extraction method or reduce it further. The Support Vector Machine is a common machine learning algorithm mostly used for classification purposes. This is because it is a fast classification algorithm, as observed by Rohit Verma et al., when compared with the Adaboost classifier. Multiclass Support Vector Machine is also highly effective in high dimensional spaces. This algorithm is also memory efficient, can separate linear inseparable data and can be used with different kernel functions as well as custom kernel functions. When compared with other classification algorithms such as K Nearest Neighbor (KNN) and random forests algorithm by Ratna Astuti Nugrahaeni et al. in "Comparative analysis of machine learning KNN, SVM, and random forests algorithm for facial expression classification", Support vector machine had the highest accuracy of 80% with the smallest amount of data. However, with a large amount of data, KNN and random forests had the highest accuracy of 98.85%.

Regarding K Nearest neighbor as a classification algorithm, M Murugappan, A M Mutawa, Sai Sruthi, Aya Hassouneh, Ali Abdulsalam, Jerrita S and Ranjana R in "Facial Expression Classification using KNN and Decision Tree Classifiers", classified six different emotions (happy, sad, anger, disgust, fear, surprise) using two different types of classifiers: K Nearest Neighbor (KNN) and Decision tree. The accuracy using K Nearest neighbor was 98.03% whilst the accuracy obtained with the decision tree was 97.21%.

Another comparison between Support Vector Machine, K Nearest Neighbor and Multilayer Perceptron classifier was done by Hivi Ismat Dino and Maiwan Bahjat Abdulrazzaq's in "Facial Expression Classification Based on SVM, KNN and MLP classifiers." The feature extraction method used here was Histogram of Oriented Gradients (HOG). The recognition rate was 93.53% when using support vector machine whilst the recognition rate was 82.97% when using MLP classifier and 79.97% when using KNN classifier.

Another classification algorithm that is usually not used for image processing is naïve bayes algorithm though it is simple, has reduced computational complexity and requires fewer amounts of training data. However, In Priyanka Nair and Subha V. 's "Facial Expression

Analysis for Distress Detection", a naïve bayes classifier was used together with a gray level difference feature extraction method. This proposed algorithm was tested on the extended Cohn Kanade (CK+) and Japanese female facial expression (JAFFE) datasets. The accuracy of this system was quite high with 98.3% accuracy for anger, 97% for Disgust, 96.1% for fear, 97.2% for happiness, 91.15% for sadness and 93.2% for surprise.

A new field that is gaining traction in recent times in the field of image processing is neural networks.

In Rabie Helaly, Mohamed Ali Hajjaji, Faouzi M'Sahli and Abdellatif Mtibaa's "Deep Convolution Neural Network Implementation for Emotion Recognition System", a facial recognition system was implemented on a raspberry pi 4 using 71 layers deep "Xception Convolutional Neural Network". Their proposed method was implemented on the FER2013 dataset and an accuracy of 94% was obtained with the NVDIA GeForce MX230 graphical processing unit, whilst an accuracy of 89% was obtained on the raspberry pi 4. The model was trained on the Graphical Processing Unit and tested on both the Graphical Processing Unit and Raspberry pi 4. On the graphical processing unit, the network was running at a speed of 5.36 frames/second whilst on the raspberry pi 4, it was running at a speed of 9.66 frames/second.

In Lutfiah Zahara, Purnawarman Musa, Eri Prasetyo Wibowo, Irwan Karim and Saiful Bahri Musa's "The Facial Emotion Recognition (FER-2013) Dataset for Prediction System of Micro-Expressions Face Using the Convolutional Neural Network (CNN) Algorithm based Raspberry Pi", a system is proposed that can predict and recognize the classification of facial emotions based on feature extraction using the Convolutional Neural Network (CNN) algorithm in real-time with the OpenCV library, namely: TensorFlow and Keras. The accuracy obtained was 65.97% in real time.

DESIGN AND REQUIREMENTS

The requirements for this facial emotion recognition device are:

- **Portability**: As stated in the introduction, one future work proposed by Baron-Cohen et al. is a longer intervention period for the emotion learning process. Hence, a portable facial emotion recognition device will ensure that a child on the autistic spectrum can learn about the facial emotions at any place and any time. This will increase the intervention period and thus, the learning curve.
- **Low cost**: This requirement ensures that the facial emotion recognition device is accessible to all families with a child on the Autism Spectrum. In developing countries such as Ghana, where Autism is still an emerging field, a device like this must be affordable so families see the need of it.
- **Low power consumption:** A low power consumption device ensures that the device lasts for a longer period with the same amount of stored energy. It also ensures that there is lower heat dissipation, which is necessary for a child-friendly device.
- **Quick time to respond:** Children on the Autism Spectrum have a short attention span; hence this facial emotion recognition device takes that into consideration. In terms of this proposed facial emotion recognition device, quick response time means the frequency at which the device can recognize the different emotions in given images is high.
- **Good computational power:** The facial emotion recognition algorithm can be quite computationally expensive. Hence, this requirement ensures that the device is well equipped to run the facial emotion recognition algorithm efficiently.

2.1 Selection of Embedded Board

Based on the above requirements, there are three hardware mediums that were considered. These include an FPGA board (Field Programmable Gate Array), a Raspberry Pi and an ATMEGA328p. A Pugh chart, as displayed below, was used to evaluate the importance of the above-mentioned embedded boards with regards to the stated requirements: portability, low cost, low power consumption, quick time to respond and good computational power.

		Option 1	Option 2	Option 3
Criteria	**Weight**	**FPGA board**	**Atmega328p**	**Raspberry Pi**
Portability	2	++	++	++
Low cost	2	-	++	+
Low power consumption	1	-	++	+
Quick time to respond	2	+	+	++
Good computational power	2	++	+	++
Total	9	5	7	7.5

Table 1: Pugh Chart for evaluating the importance of ATMEGA328p, Raspberry Pi and FPGA board using the above-mentioned requirements

As can be observed from the Pugh chart above, a field programmable gate array board was ruled out due to its high cost and high-power consumption. Both the ATMEGA328P and a

Raspberry Pi are suitable embedded boards for the facial emotion recognition process, both being quite proficient in their own regard. However, since the Atmega328p is a microcontroller, it is best suited for specific tasks. Also, one main con of the Atmega 328p is its slow clock speed of 16 MHz, as compared to the clock speed of the Raspberry Pi, which is 1.2GHz.

A Raspberry pi, on the other hand, is a microprocessor-based minicomputer; hence it is more powerful computationally and much simpler to use. Though having a good computational power, it has a reduced cost and low power consumption. A Raspberry Pi uses an operating system and hence is better for developing software application such as the facial emotion recognition algorithm. Hence, the selected embedded board is the Raspberry Pi board, specifically the Raspberry pi 4 with 2GB LPDDR4-3200 SDRAM.

The Raspberry pi 4 board has a more powerful processor compared to the previous Raspberry Pi. The processor is Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC at 1.5 GHz. In addition to a USB 2.0 port, it has a USB 3.0 port which transfers data up to ten times faster than the USB 2.0 port. The Raspberry Pi 4 board also has 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless and Bluetooth 5.0, BLE. In addition to the wireless capability, the Raspberry Pi 4 board uses gigabit ethernet. With a 2-lane MIPI DSI display port and 2-lane MIPI CSI camera port, it is easy to attach a camera and an LCD display to the Raspberry Pi 4. In terms of power, it uses 5V DC via USB-C connector with a minimum current input of 3A and this is obtained from a power adapter. The operating temperature of the board is between 0- and 50-degree Celsius ambient temperature.

2.2 Selection of Facial Expression Image Dataset

From the literature review, there are many datasets that have images showing the seven basic facial emotions: Happiness, Anger, Sadness, Fear, Disgust, Surprise and Neutrality (which is the absence of emotions). Every other complex facial emotion is a derivative of these seven basic emotions.

These datasets include the Cohn-Kanade (CK) Dataset, Japanese Female Facial Expression (JAFFE) dataset and FER2013 dataset and these are further outlined below:

- **The Japanese Female Facial Expression (JAFFE) dataset** consists of 213 images of size 256 x 256 pixels of the seven basic emotions from 10 different Japanese females. These images are 8-bit grayscale images. Unfortunately, access to this dataset is restricted to graduate school projects.
- **The Extended Cohn-Kanade (CK+) dataset** consists of 593 video sequences from 123 different individuals. The age range is from 18 to 30 years, of which 65% are female, 15% are African American and 3% are Asian or Latino. Each video shows the shift from the neutral expressions to one of the other six emotions (Happiness, Sadness, Anger, fear, disgust, sadness). The videos were recorded at a rate of 30 frames per second with a resolution of 640 x 490 pixels. Together with the JAFFE dataset, this is widely used in laboratory facial emotion recognition processes.
- **The FER2013 dataset** consists of 35,887 grayscale images of the seven basic emotions. The images are of size 48x48 pixels and is such that the face is centered and occupies relatively the same amount of space in each image. These images were not taken in a laboratory-controlled environment, like the precious datasets ,but were extracted from searches on the internet. Hence, angles and differences in expressing emotions play a key role. This dataset is presented as a csv file with the emotion

labels in one column and the pixel values in another column separated by a space. The emotion labels are represented in the csv document with numbers from 0-6, where:

Number	**Emotion label**
0	Anger
1	Disgust
2	Fear
3	Happiness
4	Sadness
5	Surprise
6	Neutrality

The dataset selected for this project is the FER2013 dataset as it contains a large number of images. Also, the difference in expression of the seven basic emotions will improve the learning curve of facial expression recognition for a child on the autism spectrum. This was one of the limitations of the “MindReading” DVD proposed by Baron-Cohen et al: The use of images in a laboratory-controlled environment with a set number of individuals produced a simplified emotion learning environment for a child on the Autism spectrum.

2.3 Selection of Facial Emotion Recognition Algorithm

Facial Emotion Recognition consists of three major stages: face recognition, feature extraction and classification. The face recognition stage ensures that features are being extracted from the right object. For images that have different objects apart from faces in the image, a face recognition algorithm will help zone in on the object under consideration, which is a face in this case. Features will then be extracted from this face. Feature extraction is very important as it reduces the computational cost during the classification stage as classification is performed on a smaller set of values that still adequately represent the image.

2.3.1 Face Recognition

The most used face recognition algorithm is the Viola-Jones face detection algorithm due to its high accuracy and fast recognition speed. This is a framework proposed by Paul Viola and Michael Jones in 2001 in their work "Rapid Object Detection using a boosted cascade of simple features." The Viola-Jones face recognition algorithm combines four main concepts: Integral Images, Haar-like features, Adaboost and the Cascade Classifier

- Representing the image as an **Integral Image** to improve the computational speed of the Haar-like features. "An Integral Image is an intermediate representation of an image where the value for location (x, y) on the integral image equals the sum of the pixels above and to the left (inclusive) of the (x, y) location on the original image."(Viola & Jones,2021)
- In a given image**, Haar-like features** take the sum of the intensities of the dark regions and subtract them from the sum of the intensities of the light region. In the Viola-Jones face detection algorithm, this is done on the integral image which increases the speed of computation of Haar-like features.

- Selecting a small number of important visual features from a larger set to yield excellent classifiers using a learning algorithm based on **Adaboost**.
- Combining these classifiers in a **cascade** which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions.

The Viola-Jones framework can be found in the OpenCV library in python and hence can be easily implemented on a Raspberry Pi board.

2.3.2 Feature Extraction

For feature extraction, there are two types of feature extraction methods: Geometry-based feature extraction methods and appearance-based feature extraction method. Geometry-based feature extraction methods use geometry to locate and extract certain key facial points such as the mouth, eyes, nose. An example of feature extraction algorithm is Active Shape Model and Active Appearance Model. Though, this form of feature extraction is more accurate than the appearance-based feature extraction, it doesn't perform well on low resolution images and unfortunately, images in the FER2013 dataset are of a low resolution. Geometry-based feature extraction methods can also be computationally expensive. Hence, an appearance-based feature extraction method will be used. The two appearance based feature extraction methods used for this project are outlined below:

- **Local Binary Pattern**

  Local Binary pattern is a texture descriptor method proposed by Timo Ojala in 2002. Local Binary pattern of a pixel is formed by selecting a center pixel's values and thresholding the 3x3 neighborhood of each pixel value. For instance, if the center value is 50, the pixel values of the 8 neighboring pixels are compared to that center value. If a particular neighboring pixel value has more than the center value, it is represented as 1. If the value is less than the center value, it is represented as 0. These binary representations are combined to get an 8-bit value. The decimal value of this 8-bit value is the local binary pattern. For values not in the center of the pixel, bilinear interpolation is used. This usually occurs with a pixel's circular neighbors. A local binary pattern is uniform when it has at most two bitwise transition from 0 to 1 or 1 to 0 and this is desired as it reduces the number of Local binary patterns in an image. This method can be found in the scikit learn package in Python

- **Histogram of Oriented Gradients**

  Histogram of oriented gradients is a feature descriptor that extracts the gradient and orientation of the edges. These orientations are calculated sections. It is best used for edge detection. HOG can be found in the scikit-learn package in python. In this algorithm, the gradient is calculated for every pixel in the image in both the x and y directions. Using the gradients, the magnitude and direction for each pixel value is calculated. Pythagoras theorem is used to find the magnitude whilst the direction is found by finding the tan of an angle. These two values: gradients and orientation are used to create the histogram. These histograms are generated for 8x8 cells of the image. These gradients are then normalized. The normalized gradients for each of the 8x8 cells are combined to form the Histogram of Oriented Gradient for the full image.

2.3.3. Classification Algorithm

There are seven machine learning algorithms used for classification purposes: Logistic regression, Naïve Bayes, Stochastic Gradient Descent, K- Nearest Neighbors, Decision trees, Random forests, and support vector machine. Amongst these, most were not applicable for the facial emotion recognition task and the requirements desired of this system. The analysis of all seven classification algorithms and their applicability/inapplicability are outlined in the table below:

Classification algorithm	**Analysis**	**Applicable/Not Applicable**
Logistic Regression	This classification algorithm is mostly used for binary classification. Though it can be employed on multiple classes using a one vs all strategy, this would be computationally expensive. In this case where there are seven different classes of emotions, a logistic regression application will require seven different models to be trained independently on the data and this might not be computationally possible on the Raspberry Pi 4	Not Applicable
Stochastic Gradient Descent	It is an optimization technique. It best works on existing linear classification algorithms like Linear Support vector machines and logistic regression. It is simple yet very efficient approach. However, it is best suited for linear data.	Not Applicable
Decision trees	Decision trees handle high dimensional data accurately and are quite time intensive. Decision trees too can easily overfit data due to the creating of over complex trees. Another disadvantages with decision trees is that biased	Not Applicable

	trees could also be created if some classes are more dominant.	
Random forests	The random forest algorithm works on the principle of the above algorithm: Decision trees. Random forests combine multiple decision trees to get a more accurate and stable prediction. Unlike the decision tree algorithm, a random forest algorithm does not usually over fit the data. However, one main limitation of random forest algorithm which makes it unsuitable for this particular process is its speed and effectiveness with real-time application. Though they are fast to train, they are quite slow when creating predictions. With this classification method, there is usually a trade off between speed and accuracy.	Not Applicable
K Nearest Neighbor	K Nearest Neighbor is one of the most used classification algorithms. It is very simple to use as well. This algorithm works on the principle that data points belonging to the same class can be found close to each other. Though this algorithm is simple to use and versatile, it is not applicable on large data sets. Another problem arises due to the hardware specification of the	Not applicable

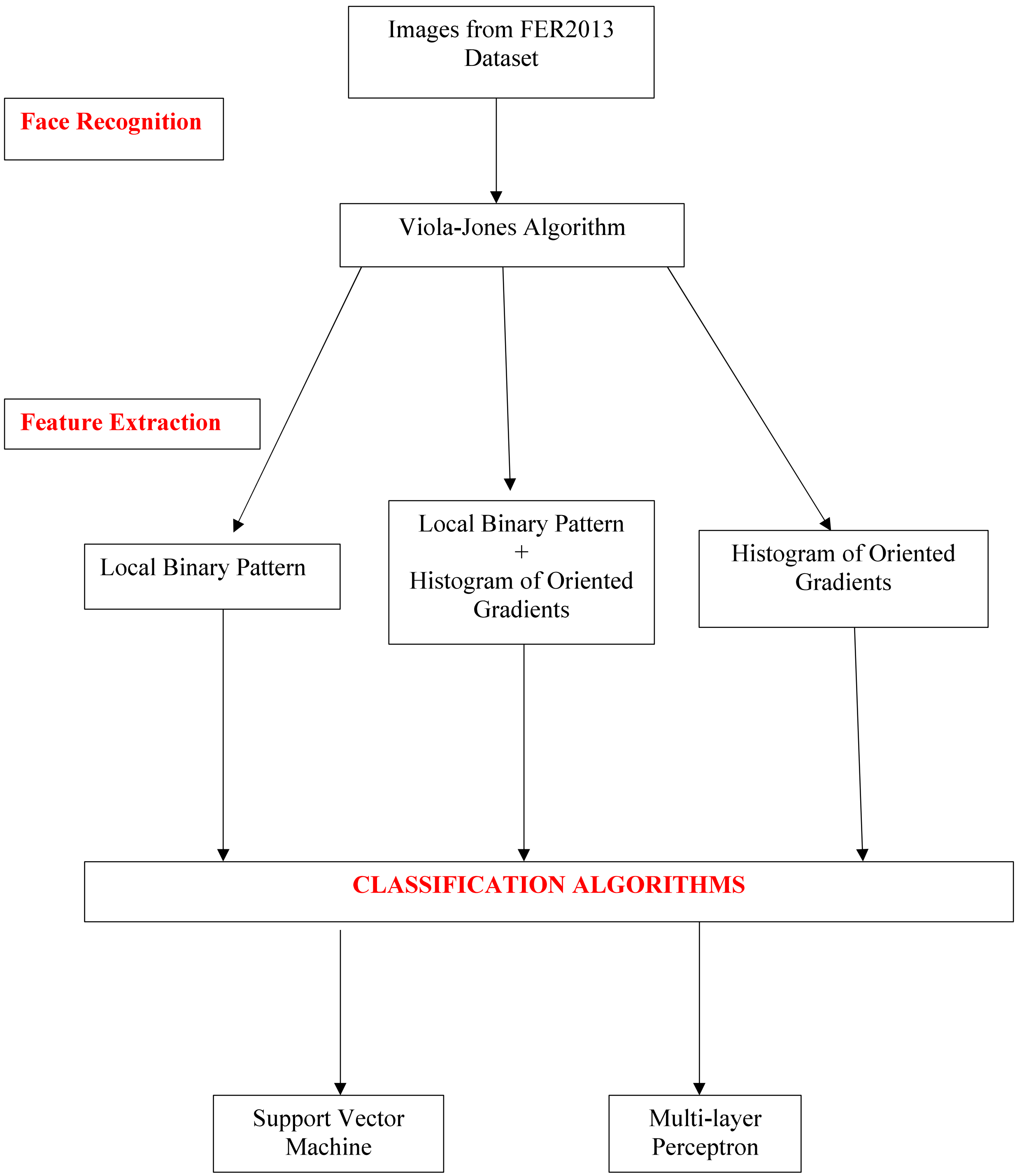
	raspberry pi board. K- Nearest neighbor is a memory-based algorithm. This means that with testing of new data, it stores all the data from the training stage in the RAM and this is memory intensive.	
Naïve Bayes	Naïve Bayes algorithm is a classification algorithm based on Bayes theorem. Bayes Theorem determines conditional probability, i.e. the probability of an event based on past conditions. With this algorithm, it assumes that the data is conditionally independent. This algorithm is extremely fast and a simple classification algorithm. However, this algorithm makes some conclusions and assumptions that might not be applicable in real life. For the Gaussian Naïve bayes, it assumes that the data from each label is drawn from a simple Gaussian distribution. Due to these assumptions, a more complex model might be more appropriate for facial expression recognition. However, naïve bayes operate well when the assumption is actually true for the dataset and when the data is high-dimensional with well-separated categories.	Not Applicable

Support Vector Machine	The final one is the Support Vector Machine which is the proposed method in this project. It is a highly accurate method with less computational power. A support vector machine consists of a hyperplane in an N dimensional space, where N = number of features, which distinctly classifies the data points. The hyperplane is selected using the maximum margin distance between the data points in the different classes. Support vector machines are also effective in high dimensional spaces. It is very versatile as different kernels can be used : linear, gaussian , polynomial or sigmoid functions. However, apart from the linear kernel, the different kernels are quite slow and will end up being killed by the Raspberry Pi board.	Applicable

Though these are the seven most used machine learning algorithms, there is another field known as Neural networks. This framework is mostly used for image processing due to its high accuracy. However, they are known to be computationally intensive. Thus, it is usually not advisable to train a neural network on a Raspberry Pi. However, the multi-layer perceptron neural network found in the scikit-learn library in python has the right computational power for the raspberry pi 4 board and thus, was implemented.

2.4. Overall Design of the System

The figure below shows the design of the facial emotion recognition device. The images from the FER2013 dataset are passed through the Viola-Jones face detection framework and then both Histogram of oriented gradient features and local binary pattern features are extracted from it. The features are both individually and collectively passed to the two different classification algorithms: support vector machine and multi-layer perceptron neural network.


Images from FER2013 Dataset
Face Recognition
Viola-Jones Algorithm
Feature Extraction
Local Binary Pattern
Local Binary Pattern + Histogram of Oriented Gradients
Histogram of Oriented Gradients
CLASSIFICATION ALGORITHMS
Support Vector Machine
Multi-layer Perceptron

RESULTS AND ANALYSIS

As stated above, two different feature extraction methods were used: Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG). Two different classification methods were used: Support Vector Machine and Multilayer Perceptron Neural Network method. Hence, there were six different combinations that were tested:

- Local Binary Patterns (LBP)+ Linear Support Vector Machine (LinearSVM)
- Local Binary Patterns (LBP) + Multilayer Perceptron Neural Network (MLP)
- Histogram of Oriented Gradients (HOG) + Linear Support Vector Machine (LinearSVM)
- Histogram of Oriented Gradients (HOG) + Multilayer Perceptron Neural Network (MLP)
- (LBP + HOG) + Linear Support Vector Machine (LinearSVM)
- (LBP + HOG) + Multilayer Perceptron Neural Network (MLP)

The number of images used for both feature extraction and classification is 13,264 images, 36.96% of the total images in the FER2013 dataset. This is because faces were detected in only these images using the Viola-Jones face recognition algorithm.

Thus, the accuracy of each of the six different methods on the 13,264 images depicting facial emotions are outlined below:

Feature Extraction	Classification	Accuracy
Local Binary Pattern	LinearSVM	32.4372%
	MLP	32.7136%
Histogram of Oriented Gradients	LinearSVM	32.3367%
	MLP	32.1608%
LBP+ HOG	LinearSVM	**35.6784%**
	MLP	34.7236%

The sections below detail the accuracy and precision for each of the seven basic emotions for the six different methods

4.2.1 Local binary patterns (LBP)+ support vector machine

Accuracy Confusion Matrix

Emotion	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral
Anger	**3.06**	0	0	75.52	0.19	3.82	17.40
Disgust	0	**0**	0	76.27	0	5.08	18.64
Fear	1.68	0	**0.63**	77.31	0	4.62	15.76
Happiness	0.83	0	0.08	**89.07**	0	1.17	8.84
Sadness	1.94	0	0	69.76	**0.22**	2.81	25.27
Surprise	2.25	0	0.9	75.45	0	**4.28**	17.11
Neutrality	0.98	0	0.12	75	0	1.35	**22.54**

Precision Confusion Matrix

Emotion	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral
Anger	**26.23**	0	0	12.55	50	19.61	13.78

Disgust	0	**0**	0	1.43	0	2.94	1.67
Fear	13.11	0	**33.33**	11.69	0	21.57	11.36
Happiness	16.39	0	11.11	**33.95**	0	13.73	16.06
Sadness	14.75	0	0	10.27	**50**	12.74	17.73
Surprise	16.39	0	44.44	10.65	0	**18.63**	11.52
Neutrality	13.11	0	11.11	19.45	0	10.78	**27.88**

Key Observations

- Though 89.05% of happy images were correctly predicted as such, 33.95% of the predicted happy images were actual happy images. This shows that most of the other emotions are misclassified as happy as shown from the accuracy table, where the happy column has the highest percentages.
- 0% of actual disgust images were predicted to be disgust. Most were predicted to be happiness with a few being predicted as Neutral or Surprise. This shows that the model could not correctly classify faces showing disgust
- 0.22% percent of actual sadness images were predicted to be sadness but however, out of the images predicted to be displaying sadness, 50% were correctly classified whilst 50% were misclassified as they depicted anger. This shows that there might be key facial expression differences between images depicting sadness in the dataset. Also, the model easily misclassifies sadness as anger
- 0.63% of fear images were correctly predicted as such but however, out of the predicted fear images, 44.44% were misclassified as Surprise. And out of the images predicted as surprise, 21.57% of the images actually depicted fear. This shows a confusion between fear and surprise in the model.

4.2.2 Local binary patterns (LBP) + multilayer perceptron neural network

Accuracy Confusion Matrix

Emotion	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral
Anger	**0.19**	0	0	60.42	0	11.28	28.10
Disgust	0	**0**	0	61.02	0	6.78	32.20
Fear	0.42	0	**0**	62.82	0	14.71	22.06
Happiness	0.08	0	0	**78.15**	0	4.17	17.60
Sadness	0	0	0	55.29	**0**	9.29	35.42
Surprise	0	0	0	63.06	0	**17.12**	19.82
Neutrality	0	0	0	59.68	0	5.02	**35.29**

Precision Confusion Matrix

Emotion	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral
Anger	**25**	0	0	12.10	0	17.20	14.38
Disgust	0	**0**	0	1.38	0	1.16	1.86
Fear	50	0	**0**	11.45	0	20.41	10.27
Happiness	25	0	0	**35.89**	0	14.58	20.64
Sadness	0	0	0	9.80	**0**	12.54	16.05
Surprise	0	0	0	10.72	0	**22.16**	8.61
Neutrality	0	0	0	18.65	0	11.95	**28.18**

Key Observations

- Though most of the images were still classified as happiness. The percentages are reduced with the highest being 63.06% of surprise images being misclassified as happiness as compared to 76.27% of disgust images being misclassified as happy in the first method of LBP and SVM. For each emotion misclassified as happiness, the percentage of images misclassified is significantly less.

- However, in this method, no image was classified as disgust, fear or sadness. For all three emotions, most of the images were misclassified as happy with a few being misclassified as surprise and neutrality.
- In addition, to these misclassifications, 0.42% of the fear images were classified as anger and out of those classified as anger, 50% of the images were fear with 25% being correctly classified as Anger and the other 25% being misclassified as they depicted happiness.
- Out of the images predicted to be happiness, surprise and neutral, the highest percentages of predicted images were the actual images.

4.2.3 Histogram of oriented gradients (HOG) + support vector machine

Accuracy Confusion Matrix

Emotion	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral
Anger	**5.93**	0	0.57	72.28	0.19	1.72	19.31
Disgust	5.08	**0**	0	74.58	0	5.08	15.25
Fear	3.78	0	**0.63**	72.06	0.63	4.41	18.49
Happiness	2.25	0	0.33	**83.82**	0.17	1.92	11.51
Sadness	4.97	0	1.08	68.68	**1.29**	2.16	21.81
Surprise	2.03	0	1.58	67.57	0	**5.63**	23.19
Neutrality	2.45	0	0.49	67.64	0.61	2.21	**26.59**

Precision Confusion Matrix

Emotion	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral
Anger	**23.66**	0	11.54	12.85	5.88	8.26	13.34
Disgust	2.29	**0**	0	1.49	0	2.75	1.18
Fear	13.74	0	**11.54**	11.67	17.65	19.27	11.62
Happiness	20.61	0	15.38	**34.18**	11.76	21.10	18.23
Sadness	17.56	0	19.23	10.82	**35.29**	9.17	13.34
Surprise	6.87	0	26.92	10.20	0	**22.94**	13.61
Neutrality	15.27	0	15.38	18.77	29.41	16.51	**28.66**

Key Observations

- Just like the Local Binary pattern feature extraction method, most images were misclassified as happy.
- No image is predicted to be disgust but most disgust images were misclassified as happy with a few being misclassified as anger, surprise and neutral.
- For anger, happiness, sadness, surprise and neutral, the highest percentage of predicted images were correctly predicted.
- Majority of predicted fear images were surprise images

4.2.4 Histogram of oriented gradient (HOG) + multilayer perceptron neural network

Accuracy Confusion Matrix

Emotion	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral
Anger	**8.03**	0	1.53	67.11	0	4.59	18.74
Disgust	3.39	**0**	0	67.79	0	5.08	23.72
Fear	4.41	0	**3.36**	65.13	0.21	9.45	17.44

Happiness	2.83	0	1.0	**79.73**	0	5.08	11.34
Sadness	8.42	0	0.86	66.09	**0.65**	5.62	18.36
Surprise	1.80	0	1.35	60.59	0	**13.06**	23.19
Neutrality	4.90	0	0.37	67.40	0.24	4.90	**22.18**

Precision Confusion Matrix

Emotion	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral
Anger	**22.58**	0	16.33	12.62	0	9.34	14
Disgust	1.08	**0**	0	1.44	0	1.17	2
Fear	11.29	0	**32.65**	11.14	16.67	17.51	11.86
Happiness	18.28	0	24.49	**34.36**	0	23.74	19.43
Sadness	20.97	0	8.16	10.99	**50**	10.12	12.14
Surprise	4.30	0	12.24	9.67	0	**22.56**	14.71
Neutrality	21.50	0	6.12	19.77	33.33	15.56	**25.86**

Key Observations

- Just like the previous methods, most emotions were wrongly classified as happy though the percentage is less than the methods with the support vector machine.
- Disgust was also not predicted as a facial expression with it being misclassified as happiness, anger, surprise or neutral.
- Apart from happy, most of the other facial expressions were also misclassified as neutral
- Though 0.65% of actual sadness images were predicted as sadness, 50% of predicted sad images were actual sad images. With fear and neutrality both also being misclassified as sadness

- Though not a high percentage , In every predicted class except for disgust and surprise, the highest percentage of predicted emotions were the actual emotions .

4.2.5 (LBP + HOG) + support vector machine

Accuracy Confusion Matrix

Emotion	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral
Anger	**11.09**	0	1.33	58.13	0.19	6.11	23.14
Disgust	6.77	**0**	0	59.32	0	8.47	25.42
Fear	4.83	0	**2.73**	63.65	0.63	8.61	19.54
Happiness	2.25	0	0.5	**82.15**	0.42	3.17	11.51
Sadness	8.64	0	1.73	56.59	**1.51**	4.53	26.99
Surprise	4.05	0	2.48	54.73	0.22	**17.11**	21.40
Neutrality	3.43	0	1.10	56.49	0.86	3.67	**34.44**

Precision Confusion Matrix

Emotion	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral
Anger	**29.29**	0	12.96	11.72	4.17	13.17	13.94
Disgust	2.02	**0**	0	1.35	0	2.06	1.73
Fear	11.62	0	**24.07**	11.69	12.5	16.87	10.71
Happiness	13.64	0	11.11	**37.99**	20.83	15.64	15.90
Sadness	20.2	0	14.81	10.10	**29.17**	8.64	14.40
Surprise	9.09	0	20.37	9.37	4.17	**31.28**	10.95
Neutrality	14.14	0	16.67	17.77	29.16	12.34	**32.37**

Key Observations

- The percentages of emotions misclassified as happy has reduced as compared to percentages obtained with the model using the individual features.
- Disgust was still not predicted with it being misclassified as happy, surprise, anger and neutral
- The most accurately classified emotion is happy
- Apart from disgust, for each predicted class, the percentage of correctly predicted images is the highest though these values are not above 50%. This is an improvement from the models obtained using the individual features.

4.2.6 (LBP + HOG) + Multilayer perceptron neural network

Accuracy Confusion Matrix

Emotion	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral
Anger	**26.00**	0	1.33	38.62	0.96	11.66	21.41
Disgust	16.95	**0**	5.08	45.76	0	10.17	22.03
Fear	17.65	0	**3.57**	40.13	0.84	20.58	17.23
Happiness	10.84	0	0.91	**67.72**	0.66	7.92	11.93
Sadness	23.54	0	2.59	34.99	**0.65**	10.58	27.65
Surprise	13.28	0	3.60	33.33	0.9	**30.63**	18.24
Neutrality	12.75	0	1.84	42.89	0.49	7.97	**34.07**

Precision Confusion Matrix

Emotion	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral

Anger	**21.52**	0	8.64	10.68	17.86	11.96	13.38
Disgust	1.58	**0**	3.7	1.42	0	1.17	1.55
Fear	13.29	0	**20.98**	10.09	14.29	19.21	9.80
Happiness	20.57	0	13.58	**42.92**	28.57	18.62	17.08
Sadness	17.25	0	14.81	8.56	**10.71**	9.60	15.29
Surprise	9.33	0	19.75	7.82	14.29	**26.67**	9.68
Neutrality	16.45	0	18.52	18.49	14.29	12.74	**33.21**

Key observations

- Though the percentage of images misclassified as happy has reduced drastically. The accuracy of the happy images has also reduced from 82.15% in the method using the fusion of LBP and HOG features with a Linear Support Vector Machine to 67.72% in this method.
- Images showing a disgust facial expression are still not classified.
- The classification accuracy of anger images has increased from 11.09% to 26%
- The classification accuracy of surprise images also has increased from 17.11% to 30.63%
- However, in addition to happy, the classification accuracy of sadness and neutral has reduced, though not by a huge amount.
- Except for images depicting sadness and disgust, the highest percentage in every predicted class is images from the actual class.

CONCLUSION

The fusion of Local Binary Patterns and Histogram of Oriented Gradients features produced a higher accuracy than the methods using individual features. Thus, this is the desired feature extraction method together with a support vector machine classifier. Though, the multilayer perceptron neural network increased the accuracy of the other emotions apart from happy namely: anger, surprise and fear, the overall accuracy of the method with the Linear support vector machine was higher. Secondly, the Multilayer perceptron neural network takes more time to process than the support vector machine.

Hence, in conclusion, a facial emotion detection device is proposed in this paper on a raspberry pi 4 using a fusion of Local Binary Patterns and Histogram of Oriented Gradients together with a linear support vector machine. This system produced an overall accuracy of 35.68%. The python code for the system can be found in the appendix below.

LIMITATIONS AND FUTURE WORKS

As can be observed from above, the overall accuracy of the system is less than 50% and this can be attributed to the following factors:

- The dataset is not balanced, i.e., there are not equal number of images in each emotion class of the dataset. For future works, the model dataset can be balanced out by using only equal numbers of images for each class of emotion.
- Images in the FER2013 dataset were not taken in a laboratory-controlled environment, hence there might be key differences in the display of the same facial expression. Though this was intended, it contributed to the low accuracy of the system.
- Due to hardware restraints, the support vector machine was unable to be run with other kernels: polynomial, rbf or sigmoid, which would have been more adapted to

the dataset. For future works, the model can be trained on a computer with one of the above-mentioned kernels and implemented on the Raspberry Pi.

- Also, due to hardware restraints, a more complex version of the neural network could not be trained on the raspberry pi. However, for future works, complex form of a neural network, possibly one with back propagation will be trained on a computer and implemented on the raspberry pi.

REFERENCES

B. Islam, F. Mahmud and A. Hossain, "High Performance Facial Expression Recognition System Using Facial Region Segmentation, Fusion of HOG & LBP Features and Multiclass SVM," 2018 10th International Conference on Electrical and Computer Engineering (ICECE), 2018, pp. 42-45, doi: 10.1109/ICECE.2018.8636780.

Baron-Cohen, S., Golan, O., & Ashwin, E. (2009). Can Emotion Recognition Be Taught to Children with Autism Spectrum Conditions? *Philosophical Transactions: Biological Sciences,364*(1535), 3567-3574. Retrieved April 25, 2021, from http://www.jstor.org/stable/40538150

Caifeng Shan, Shaogang Gong and P. W. McOwan, "Robust facial expression recognition using local binary patterns," IEEE International Conference on Image Processing 2005, 2005, pp. II-370, doi: 10.1109/ICIP.2005.1530069.

Emotional development in children with autism spectrum disorder. Raising Children Network. (2020). Retrieved 6 October 2020, from https://raisingchildren.net.au/autism/development/social-emotional-development/emotional-development-asd.

H. I. Dino and M. B. Abdulrazzaq, "Facial Expression Classification Based on SVM, KNN and MLP Classifiers," 2019 International Conference on Advanced Science and Engineering (ICOASE), 2019, pp. 70-75, doi: 10.1109/ICOASE.2019.8723728.

K. Meena and A. Suruliandi, "Local binary patterns and its variants for face recognition," 2011 International Conference on Recent Trends in Information Technology (ICRTIT), 2011, pp. 782-786, doi: 10.1109/ICRTIT.2011.5972286.

Kanner, L. (1943). Autistic disturbances of affective contact. *Nervous Child, 2,* 217–250.

L. Zahara, P. Musa, E. Prasetyo Wibowo, I. Karim and S. Bahri Musa, "The Facial Emotion Recognition (FER-2013) Dataset for Prediction System of Micro-Expressions Face Using the Convolutional Neural Network (CNN) Algorithm based Raspberry Pi," 2020 Fifth International Conference on Informatics and Computing (ICIC), 2020, pp. 1-9, doi: 10.1109/ICIC50835.2020.9288560.

M. Murugappan et al., "Facial Expression Classification using KNN and Decision Tree Classifiers," 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP), 2020, pp. 1-6, doi: 10.1109/ICCCSP49186.2020.9315234.

Ofori-Atta, A., & Ohene, S. (2014). *Changing Trends in Mental Health Care and Research in Ghana* (pp. 69-78). Sub-Saharan Publishers.

P. Kumar, S. L. Happy and A. Routray, "A real-time robust facial expression recognition system using HOG features," 2016 International Conference on Computing, Analytics and Security Trends (CAST), 2016, pp. 289-293, doi: 10.1109/CAST.2016.7914982.

P. Nair and S. V., "Facial Expression Analysis for Distress Detection," 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2018, pp. 1652-1655, doi: 10.1109/ICECA.2018.8474761.

R. Verma and M. Dabbagh, "Fast facial expression recognition based on local binary patterns," *2013 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, Regina, SK, Canada, 2013, pp. 1-4, doi: 10.1109/CCECE.2013.6567728.

R. A. Nugrahaeni and K. Mutijarsa, "Comparative analysis of machine learning KNN, SVM, and random forests algorithm for facial expression classification," 2016 International Seminar on Application for Technology of Information and Communication (ISemantic), 2016, pp. 163-168, doi: 10.1109/ISEMANTIC.2016.7873831.

R. Helaly, M. A. Hajjaji, F. M'Sahli and A. Mtibaa, "Deep Convolution Neural Network Implementation for Emotion Recognition System," 2020 20th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), 2020, pp. 261-265, doi: 10.1109/STA50679.2020.9329302.

Raspberry Pi(2021). Retrieved 26 April 2021, from https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/

Rump, K., Minshew, N., Giovannelli, J., & Strauss, M. (2009). The Development of Emotion Recognition in Individuals with Autism. *Child Development, 80*(5), 1434-1447. Retrieved April 25, 2021, from http://www.jstor.org/stable/25592082

S. Liao, W. Fan, A. C. S. Chung and D. Yeung, "Facial Expression Recognition using Advanced Local Binary Patterns, Tsallis Entropies and Global Appearance Features," 2006 International Conference on Image Processing, Atlanta, GA, USA, 2006, pp. 665-668, doi: 10.1109/ICIP.2006.312418.

S. L. Happy and A. Routray, "Robust facial expression classification using shape and appearance features," 2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR), 2015, pp. 1-5, doi: 10.1109/ICAPR.2015.7050661.

S. Roy Supta, M. Rifath Sahriar, M. G. Rashed, D. Das and R. Yasmin, "An Effective Facial Expression Recognition System," 2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE), 2020, pp. 66-69, doi: 10.1109/WIECON-ECE52138.2020.9397965.

Suchitra, Suja P. and S. Tripathi, "Real-time emotion recognition from facial images using Raspberry Pi II," 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2016, pp. 666-670, doi: 10.1109/SPIN.2016.7566780.

P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517.

What Is Autism? | Autism Speaks. Autism Speaks. (2020). Retrieved 5 October 2020, from https://www.autismspeaks.org/what-autism.

APPENDIX

```
from skimage.feature import local_binary_pattern
from sklearn.svm import LinearSVC
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import math
from sklearn.model_selection import train_test_split
import sklearn.metrics as metrics
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neural_network import MLPClassifier
from skimage.feature import hog
import cv2

# settings for Local Binary Patterns
radius = 3
n_points = 24


training_images = pd.read_csv('fer2013.csv')
data = []
labels = []
eps = 1e-7

# Initializing Linear Support Vector Machine and MLPCLassifier
clf = LinearSVC( random_state=0,tol=1e-5
mlp = MLPClassifier(hidden_layer_sizes=(8,8,8), activation = 'relu', solver = 'adam',max_iter = 500)


width,height = 48,48
pixels_list = training_images['pixels'].tolist()
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
```

```
for pixel_item in pixels_list:
    face = [int(pixel) for pixel in pixel_item.split(' ')]
    face = np.asarray(face,dtype = 'uint8').reshape(width,height)

    # Viola-Jones face recognition algorithm
    faces = face_cascade.detectMultiScale(face)
    if len(faces) ==  0:
       continue
    for (x,y,w,h) in faces:
        roi_face = np.asarray(face[y:y+h, x:x+w])
        roi = cv2.resize(roi_face,(40,40),interpolation = cv2.INTER_AREA)

    #Computing Histogram of Gradients for the detected face in the image
    facialhog =  hog(roi, orientations =8, pixels_per_cell = (16,16), cells_per_block =(1,1), feature_vector=True)

    #Computing Local Binary Patterns for the detected face in the image
    faciallbp = local_binary_pattern(roi, n_points, radius, method = 'uniform')
    (hist, _)= np.histogram(faciallbp.ravel(),bins=np.arange(0,n_points+3), range =(0, n_points +2))
    hist = hist.astype("float")
    hist /= (hist.sum() + eps)

    individual_data= np.hstack([hist, facialhog])
    data.append(individual_data) #Append  the fusion of HOG+LBP features
    #data.append(facialhog)      #Append HOG features
    #data.append(hist)           #Append LBP features

    labels.append(training_images['emotion'].iloc[pixels_list.index(pixel_item)])

    print('Percentage read:',(pixels_list.index(pixel_item)/35886) * 100)

print (len(data))
print (len(labels))
```

```
x_train, x_test, y_train, y_test = train_test_split(data,labels,test_size = 0.30, random_state=40)
print("Done splitting data")
training_data = np.array(x_train,dtype = object)
print("Training data array")
training_labels = np.array(y_train)
print("Training labels array")
#clf.fit(training_data, training_labels)    #Training LinearSVM
mlp.fit(training_data, training_labels)     #Training Multilayer Perceptron Neural Network
print("Done fitting model")

y_test = np.array(y_test

#prediction_data = clf.predict(x_test)      #Testing LinearSVM
prediction_data = mlp.predict(x_test)       #Testing Multilayer Perceptron Neural Network
print("Prediction data array")

# Metrics
print("Accuracy:", metrics.accuracy_score(y_test, prediction_data))
print("Precision:",metrics.precision_score(y_test, prediction_data,average='micro'))
print("Recall:",metrics.recall_score(y_test,prediction_data,average='micro'))
print( metrics.confusion_matrix(y_test,prediction_data,normalize='true'))
print(metrics.confusion_matrix(y_test, prediction_data, normalize ='pred'))
```