An Empirical Comparison of Face Emotion Recognition Algorithms: Support Vector Machines and Convolutional Neural Network

O'Connor, Stephen
Computer Science
Syracuse Universtiy
Flordia, United States of America
Soconn02@syr.edu

Parekh, Abhishek
Cybersecurity
Syracuse Universtiy
Toronto, Ontario, Canada
apparekh@syr.edu

Facial Emotion Recognition is a technique that analyses and recognizes features from the human face that are preprocessed from images or videos, conducted using machine learning algorithms. Face recognition falls under biometrics since it verifies and recognizes an individual for the purpose of authentication. The paper mainly focusses on two machine learning algorithms used for sentimental analysis using facial features. The first algorithm is a Deep neutral network, and the second implementation is Support Vector Machines. The models are trained for the detection and classification of facial features and later tested to see how well the models performed and where improvements can be made. This paper investigated the performance of a deep neural network and two SVM classifiers, not to mention that both the implementations were trained and tested using the FER2013 dataset. The results showed that the Polynomial function used as the SVM classifier had an accuracy of 40.6% and a precision score of 40.2%. While on the other hand, Radial Basis function used as the second SVM classifier had an accuracy of 41.3% and a precision score of 42%. Similarly, the neural network was trained, tested and ran for 100 iterations, once it surpasses the 15th iteration, the model accuracy hovers around 50%. From the above findings, the models can detect facial expressions, however the models presented in this paper are not well suited for large datasets, the performance will improve as the size of the dataset decreases. Although, to improve the performance on the FER2013 dataset, we need to enhance the image preprocessing phase and work towards better handling the image data, so that it efficiently maps vital facial features into the training model. Moreover, face recognition is used for security, criminal enforcement, authentication, and marketing. Due to its usages, there are several concerns regarding the use of pictures since it highlights the issue of privacy and the misuse of images. We need to be careful when utilizing images of people, we need to make sure that the image dataset is from a credible source that had the permission to capture images from a variety of people. The use cases for facial recognition are immense and we need to make sure that we gain the benefits whilst handling privacy concerns.

Keywords— Facial Emotion Recognition (FER) Algorithm, Emotion Recognition Algorithms, Sentiment Analysis, the Facial Action Coding System (FACS), Support Vector Machines, Kernel Function, Hyperparameter tuning, training dataset, testing dataset, Cross Validation, Feature Extraction, Model Evaluation, Accuracy, Precision, Recall, Classification report, Confusion Matrix, Principal Component Analysis (PCA), Radial Basis Function, Polynomial Function

I. Introduction

Over the previous 20 years, there has been work within Computer Science Academia and the private sector to develop algorithms and applications that recognize human emotions based on facial imagery [1]. The basis for the use of facial imagery is that humans display their current emotional state through a set of common facial expressions. These common sets of facial expressions are used subconsciously by humans during social interactions with other humans and other mammals [1]. The study of the universal facial expressions on varying cultures and languages can be traced back to Charles Darwin's 'The Expression of the Emotions in Man and Animals' in 1872 [1]. From there several journal articles were published over the years on the topic, with some of the most cited articles being 'Facial motion in the perception of faces and of emotional expression.' (Bassili, 1978) and 'Facial Expressions of Emotion: An Old Controversy and New Finding' (Ekman, 1992) [1]. Articles continue to be published on the topic with a more modern example being Barrett's Study in 2021 'AI weighs in on debate about universal facial expressions [1]. Over the years there has been a consensus that there are Seven Universal Facial Expressions of Emotion that humans share: happiness, surprise, contempt, sadness, fear, disgust, and anger [1].

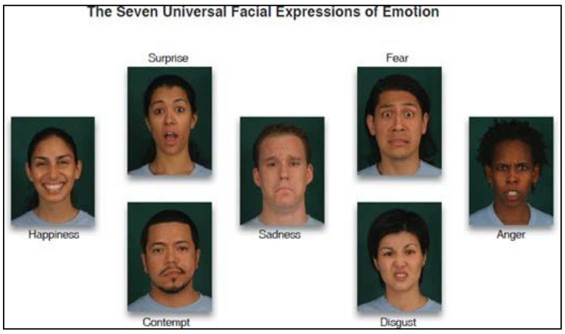

Fig. 1. The Seven Universal Facial Expressions of Emotion as identifed by the Federal Bureau of Investigatoin

Given a human is not suppressing or misrepresenting their expressions then a computer application should be able to identify the human's instantaneous emotional state using image recognition processing on the human's face. Based on reviewing journal articles over the last 5+ year period (end of the 2010s and start of the 2020s) many Face Emotion Recognition Algorithms rely on Artificial Intelligence (AI) and Machine Learning (ML) Algorithms. With the design and implementation of these ranging from linear regression to complex neural networks. Some examples of these models include Convolutional Neural Network, Deep Learning, and Support Vector Machines (SVM).

Algorithm 1 will use an SVM model for the purpose of classifying and detecting expressions from images. We first prepare our datasets, by splitting it into two portions; for training the algorithm and the other dataset for testing the algorithm, through this we get a clear idea of how well our algorithm works and where it needs to be improved. Then, we choose the kernel function, this step is conducted for transforming the data into a higher or lower dimensionality space, this aids in easy handling of the key features within the images. Later, we can train the SVM on the training dataset and our chosen kernel function that we prepared. During the training process, the SVM learns key features and values of the hyperplane and the support vectors. Once the model has been trained, we would like to add more optimization to the model, for this we can adjust the hyperparameters and attributes within the kernel to achieve the optimal performance for the model. Our last step would be to test and evaluate the performance of our trained model, for this we can use the test dataset and run the model on it to see how accurate its performance, this we can calculate metrics such as precision, accuracy, and recall; the test dataset will include data that model has never seen before.

Algorithm 2 is a Convolutional Neural Network (CNN) approach to the Face Emotion Recognition Detection problem. To summarize a CNN is a Deep Learning algorithm that accepts input images and defines weights and biases to aspects (objects and features) within the image; these items are used to differentiate the different emotions from the image. A CNN is broken into multiple parts including the Preprocessing, Convolution Layer, Pooling Layer, and the Classification Layer (or the Fully Connected Layer). [5] The CNN will be created using Python in-conjunction with opensource software including but not limited to TensorFlow, and OpenCV.

The two algorithms will be trained using an identical Training Data Set then tested using identical Testing and Validation data sets. Accuracy, Precision, and other metrics will be calculated and reviewed for each algorithm's results. From there the team will analyze the metrics to highlight the Positives and Negatives of each algorithm. These results will be presented in a confusion matrix in Section IV Results. If any conclusions can be obtained from the results, they will be stated in this paper's conclusion.

II. Previous Works

A. Emotion Detection with Facial Feature Recognition Using CNN & OpenCV

Engineers out of the Apex Institute of Technology at Chandigarh University in Mohali, India developed an algorithm that allows the computer to identify human emotion using a Convolution Neural Network (CNN) and OpenCV; The team referred to their algorithm as EDR. These results were presented at the 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE). The algorithm was developed using existing open-source Deep Learning Python Libraries; the libraries include but are not limited to loadImg, ImgToArray, Dense, Input, Dropout, GlobalAveragePooling2D, Flatten, Conv2D, Batch Normalization, Activation, an MaxPooling2D. The Algorithm used a multi-layer CNN. The team trained their algorithm with pictures (from Kaggle) that had the Emotion previously defined within the backend. These images were converted to Grayscale images prior to being fed to the Algorithm. The journal from the conference indicates an accuracy of 80% based on data from a camera sensor. [3]

B. Emotion Recognition Using Facial Expressions

Engineers out of the University of Warsaw presented the results of their training and testing of dualling (k-Nearest Neighbors (k-NN) Classifier and Multilayer Perceptron MLP) emotion recognition algorithms at the International Conference on Computational Science in Zurich Switzerland. The team used a Kinect to put data into their emotion recognition algorithms. The Kinect was chosen for its 3D face modeling capability due to its ability to capture high-rate images with visible light and infrared images along with IR data for distancing. This combination of data enables the creation of a 3D Model; the model was based on specific spatial coordinate points on the face. The changes in spatial coordinates are identified as Action Units (AU) from the Ekman and Friesen FACS System. From there the team fed 6 specific Action Units into their Recognition Algorithms to determine one of the seven universal face expressions. [4]

The team's emotional automatic recognition algorithms using AU have conducted k-NN and MLP classifiers. The k-NN nearest neighbor classifier is an algorithm where the input is classified by a plurality vote of its neighbors, where the output is assigned to the class most common among its k nearest neighbors. Where if k = 1, then the answer is the nearest neighbor, and the team chose a k of 3. The MLP Classifier is an algorithm that is consistent of 3 layers (Input, Hidden, Output) of neurons that feed information to the next other neurons leading to an output. The team configured the MLP with 7 neurons in the hidden layer. [4]

The team captured six live human male subjects depicting the different emotions with the Microsoft Kinect. Each Subject was an identical distance from the camera and kept the same head orientation; varying in distance or head orientation could significantly change the coefficients of the 6 key Action Units. Each subject displayed each of the 7 emotions 6 times (2 Sessions X 3 Trials) resulting in 252 different facial expressions. [4]

Note: The algorithm required 3D video clips as input data instead of static 2D image used within this project.

C. Facial Emotion Detection Using Convolutional Neural Networks

Hosur submitted his research on using convolution neural network for Facial Emotion Recognition (FER) to the 2022 IEEE (Institute of Electrical and Electronics Engineers) 2nd Mysore Subsection International Conference (Mysuru Conn). The algorithm was trained and tested using premade data fed in a Web Camera interface. The research broke the algorithm into 2 distinct parts: Removal of Background from Image and Facial Feature Extraction. The feature extraction is done by outlining points from around the face to generate a primary expression vector. Once the Facial Image was extracted, the facial features were identified using filtering and augmentation. The Features were then input into the model for classification. The Model was a 2 Level Model with a total network size of 22 convolution layers. In the end, Hosur claimed a 96% accuracy rate. [6] Note: Hosur indicated the same data set was used to train and test the data which may explain the high percentage as compared to the other studies identified with Section 3 Previous Works.

D. Automatic Human Emotion Recognition System using Facial Expressions with Convolution Neural Network

Engineers out of India produced a research paper on an Automated Facial Recognition System for the International Conference on Electronics, Communications, and Aerospace Technology Conference. The paper processed a system that used a large Database of previously classified Face images to train the Model. Their algorithm used High Boost filtering to reduce the noise in the image. The logic behind the high-boost filtering was that it preserved the low-frequency components during the filtering process. Feature Extraction was done on the filtered image and used to classify images within a CNN. In the end the best algorithm was able to detect the emotion from the image with 91% accuracy. [7]

E. Real and Fake emotion detection using Enhanced Boosted Support Vector Machine Algorithm

During the era of deep fakes, we can encounter images with real and fake emotions, it may be difficult to categorize solely with the human eyes. In addition, some models may even deem certain fake emotions to be real since it has not been taught to catch and intercept fake emotions. In addition, not much research and development has been done to catch fakes, the average accuracy of the models ranges between 51% - 76%. The paper proposes an algorithm called an Enhanced Boosted Support Vector Machine (EBSVM) [8]. EBSVM is a cutting-edge method for identifying critical thresholds needed to comprehend artificial emotions. In addition to SASE-FE, they have developed a new dataset called FED that includes 50 individuals' actual and artificial emotion photos [8]. Every time an iteration of the EBSVM is performed, the ensemble classifier is used to classify the full set of data [8].

F. Unknown Object Detection Using a One-Class Support Vector Machine for a Cloud-Robot System

Running multiple robots in an indoor warehouse while they communicate with each other can be tremendously taxing in terms of computational power. This paper investigates a solution that runs multiple robots on a cloud-based system. Effective indoor service robot deployment requires unrestricted mobility, improved robot view, and a rigorous categorization of objects and barriers using vision sensor data [9]. The paper proposes the solution of implementing a support vector data description based one-class support vector machine for the detection of unknown objects [9]. To train and identify objects in the environment, a cloud-based convolutional neural network (CNN) model with a SoftMax classifier is employed [9]. To train the model a training dataset was set up using an object image dataset from an open-source repository and captured images of the lab environment. The proposed solution showed great results regarding the detection of unknown objects and identification, its performance was compared with other algorithms, which it outperformed. The accuracy performance of the object classification model was 97.67% [9].

G. Classification of Malicious Android Applications using Naïve Bayes and Support Vector Machine Algorithms

Malicious software is always the concern for any operating system and as more users start using a particular OS, hackers become determined to develop stronger and powerful malware embedded within software applications that go unnoticed for a prolonged period. A solution proposed by Yilmaz, Taspinar and Koklu was to develop a classification system capable of detecting Malicious software [10]. A dataset was prepared, containing 2854 malicious applications and 2870 harmless applications [10]. A SVM model and a Naives Bayes (NB) model were trained using features extracted from the dataset, mainly being the signatures of a given program, and it would match the signatures on its database, a match indicates a harmless application [10]. According to the metrics, the SVM model had a success rate of 90.9%, whereas the NB model had a success rate of 92.4% [10].

H. A Hybrid Approach Based on Principal Component Analysis for Power Quality Event Classification Using Support Vector Machines

Power quality is a concept within the field of electrical engineering, it is known as the ability of a power grid to supply a clean stable flow of electricity without any interference or power breaks [11]. The supply of electricity is an essential need for any community, an interruption in the supply causes massive inconvenience to numerous businesses and households. The paper proposes that a classification be used to detect faults in power quality events [11]. Using Hilbert transform and Wavelet transform, key features are extracted to train the model [11]. The use of PCA is committed to choosing between the most crucial features from the data [11]. As a result, the SVM model that utilized both signal-processing techniques delivered the best accuracy and precision results [11].

III. Experiment Design

A. Training Data Sets and Test Data Sets

The algorithms used in this paper will require learning and refining their precision when running on a sample of data. For this purpose, we will split the entire dataset into two portions, one part will consist of data used for training the models, and the other portions will consist of data for testing the model [12]. The split ratio will be around 80% for the training dataset and 20% for the testing dataset. The data from the training dataset is the one that the models will extract features from, this is done since the model will see this for the first time, and it will help it keep the features and classification instructions when it sees brand new data [12]. The testing dataset provides the model with the chance to see how well it will perform. This is an authentic and unbiased procedure that gives us the opportunity to improve our models where necessary. This paper will use the FER-2013 Data set from the Kaggle community. This dataset consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that they are centered and occupy about the same amount of space in each image.

Note that the datasets will not include any imagery with face coverings or masks as successful emotion recognition will be challenging. study out of the Department of Psychiatry and Psychotherapy, University of Tubingen, Tubingen, Germany concluded "wearing a face mask affects social interaction in terms of impaired emotion recognition accuracy and altered perception of threat." (Grahlow et al., 2022) This study was done using humans trying to identify the current emotion state using images with varying masks artificially placed on the image. With this study in mind, the experiment's design chose not to include any faces with coverings. [2]

B. *Test Execution and Parameters*

The test will be conducted over the testing dataset, which we reserved for later use. This will be an unbiased evaluation of our models since this data is foreign and never seen by the model; providing us a clear picture of how well the model is made and how well it will perform in real-life situations [12]. Once the test has been conducted, we can create a confusion matrix from the results, and calculate the accuracy and the precision. This will lead us to decide which parameters we need to change to further improve the performance of the models, such as for SVM(s)we can change the kernel function, or the support vectors themselves. In the neural network model, we can change the weights, or how the features are extracted in the image preprocessing step. Furthermore, having cropped images of the face alone will provide better results, compared to having images of a person with a large background, this is since the preprocessor will need to clear out any unnecessary clutter to gain crucial features from the images, for our case, we only need features from the human face.

IV. Results

A. *Algoritm 1 Implementation and Results*

The SVM approach uses a supervised learning method, where the data is labeled to train the algorithm to classify the data and then evaluate the outcome based on what it learned. We must first prepare the training dataset and the testing dataset; this will be done by splitting the original dataset. Once the training set has been prepared, we must preprocess the data, this will include applying a PCA module to reduce the dimensionality in the dataset, a face cropper to gain the essential portions of the image and finding the centroids for the purpose of matching and predicting. Afterward, a kernel function will be used, this aids in classifying the data based on its similarities and differences. Based on the steps conducted, we can train the SVM model, therefore it can learn and retain the information regarding the classification. The data preprocessing steps can be found in Fig. 1.

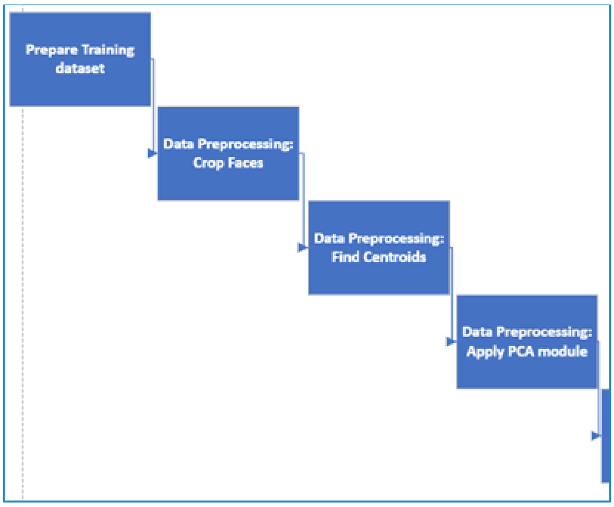

Fig. 2. Data Preprocessing steps for the SVM model

Additionally, the trained model will be evaluated using the test dataset, through this the model will be judged on its performance and accuracy in predicting the outcome. If any erroneous predictions are made it will be reflected in the confusion matrix and the calculations regarding its precision, to resolve this we can tune the hyperparameters of the model and use a different kernel function, resulting in an optimized model; it will gain better performance and accuracy. The implementation and evaluation of the SVM model can be seen in Fig. 2.

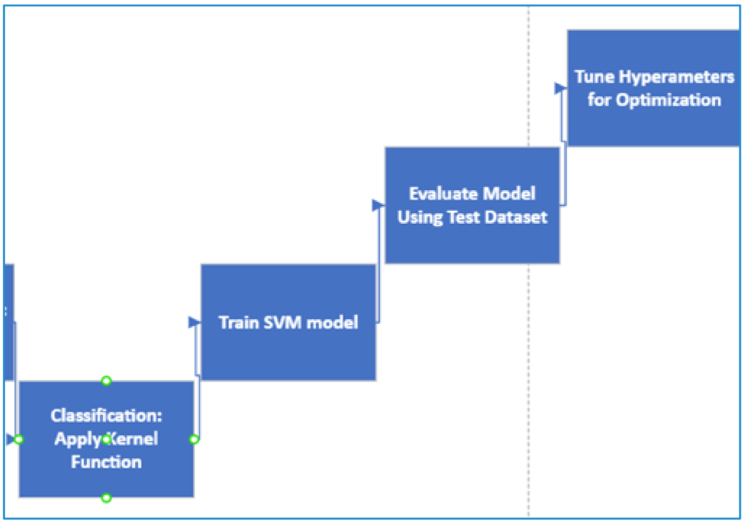

Fig. 3. The Implementation and testing of the SVM Model

For building the SVM model using the FER2013 dataset, the labels from each image category were extracted and stored in a NumPy array, whilst the images themselves were also stored in a separate NumPy array. Image preprocessing steps were applied to the images, these include flattening the image and applying a PCA module. The PCA module is used to transform the image dataset to a lower dimensionality space, this enables it to retain the essential features needed for training the SVM model, at the same time omitting any necessary noise in the images. As shown in Fig. 4. we see that the dimensionality of the input array containing the images has been reduced from (28709, 2304) to (28709, 100).

Furthermore, once the image preprocessing step had completed, the training dataset was used to train two separate SVM models with their very own classifier. The two classifiers used for this paper are polynomial function and radial basis function, both kernel functions were set under the same conditions and used the same training and testing datasets to evaluate their performances. Once each classifier was trained and evaluated, their accuracy, precision, recall and F1 scores were calculated in order to illustrate how well each classifier performed in categorizing facial expressions. The polynomial kernel function had an accuracy of 40.6%, while its precision was 40.2%. Whereas the radial basis function had an accuracy of 41.3% and a precision score of 42.05%, the metric calculations are shown in fig. 5 and fig. 8 for each classifier, respectively. Additionally, from the bar charts shown in fig. 11 and fig 12. displaying the metrics results for the polynomial function and the radial basis function, respectively, we see that the radial basis function is slightly better than the polynomial function, however, it is not an impression difference in performance, hence choosing either classifier would lead to very similar results.

```
(28709, 2304)
(28709, 100)
[[ 1.10614879e+03  2.25958130e+03 -2.97102509e+02 ...  9.16064570e+01
   3.14547709e+01  7.45050387e+01]
 [-2.11039876e+03  3.09018887e+02 -1.57002282e+02 ... -1.32136625e+02
  -6.51018048e+01 -2.54726389e+01]
 [ 1.41384579e+03 -1.47415078e+03 -1.31843002e+02 ... -3.12174162e+01
   5.88332625e+01 -8.19439649e+01]
 ...
 [ 1.84367155e+03 -4.99959387e+02 -2.66297255e+03 ... -4.37009824e+01
   3.49809851e+01 -2.96513653e+00]
 [ 3.12062321e+03 -3.32438991e+02 -7.28045375e+02 ...  7.94407500e+01
   1.11235265e+01 -3.15300040e+00]
 [-1.31229329e+03  2.90620389e+02 -1.26595629e+03 ... -1.23826941e+02
   2.42023041e+00 -2.00575023e+02]]
```

Fig. 4. Applying PCA Module for image preprocessing

```
Accuracy with polynomial Kernel: 40.61302681992337 %
Precision with polynomial Kernel: 40.200140426420816 %
Recall with polynomial Kernel: 40.61302681992337 %
F1 with polynomial Kernel: 40.086872931597206 %
```

Fig. 5. Metrics for Polynomial classifier in the SVM Model

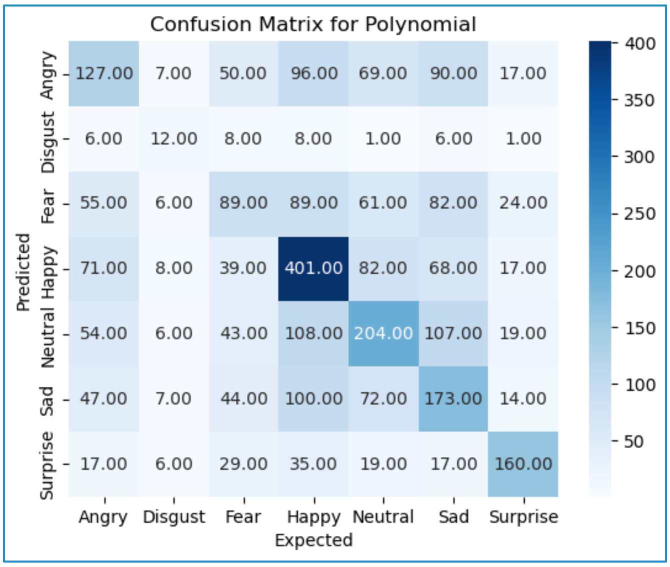

Fig. 6. Confusion Matrix for Polynomial Classifier

	precision	recall	f1-score	support
angry	0.34	0.28	0.30	456
disgust	0.23	0.29	0.26	42
fear	0.29	0.22	0.25	406
happy	0.48	0.58	0.53	686
neutral	0.40	0.38	0.39	541
sad	0.32	0.38	0.35	457
surprise	0.63	0.57	0.60	283
accuracy			0.41	2871
macro avg	0.39	0.38	0.38	2871
weighted avg	0.40	0.41	0.40	2871

Fig. 7. A comprehensive classification report for Polynomial Classifier used in the SVM model

```
Accuracy with Radial basis kernel: 41.344479275513756 %
Precision with Radial basis kernel: 42.05698435123828 %
Recall with Radial basis kernel: 41.344479275513756 %
F1 with Radial basis kernel: 39.03964506294846 6 %
```

Fig. 8. Metrics for Radial Basis classifier in the SVM Model

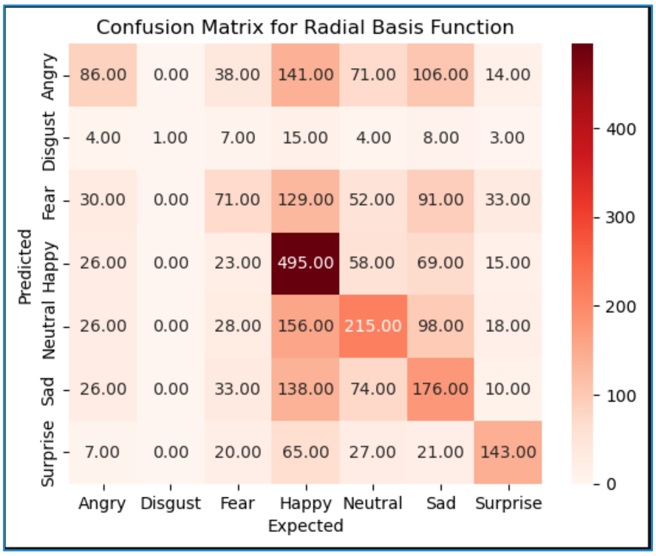

Fig. 9. Confusion Matrix for Radial Basis Classifier

	precision	recall	f1-score	support
angry	0.42	0.19	0.26	456
disgust	1.00	0.02	0.05	42
fear	0.32	0.17	0.23	406
happy	0.43	0.72	0.54	686
neutral	0.43	0.40	0.41	541
sad	0.31	0.39	0.34	457
surprise	0.61	0.51	0.55	283
accuracy			0.41	2871
macro avg	0.50	0.34	0.34	2871
weighted avg	0.42	0.41	0.39	2871

Fig. 10. A comprehensive classification report for Radial Basis Classifier used in the SVM model

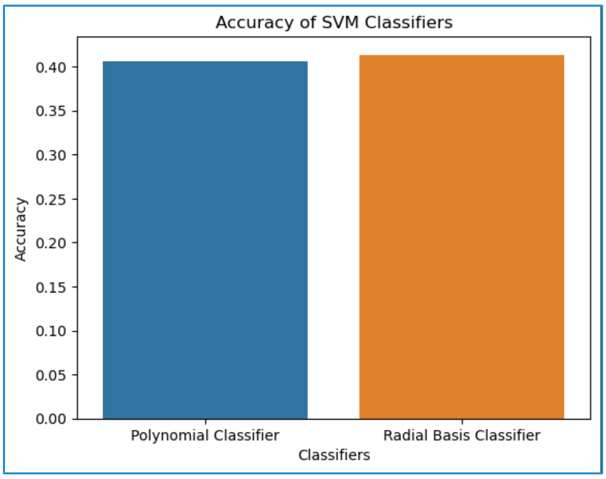

Fig. 11. Bar graph displaying the accuracy of the two SVM classifiers used in this paper

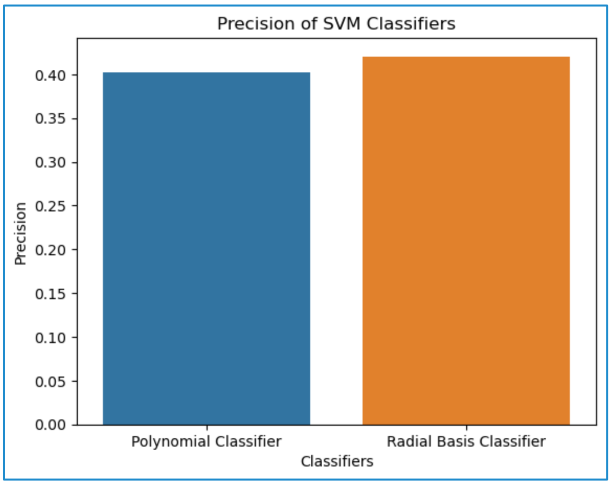

Fig. 12. Bar graph displaying the precision of the two SVM classifiers used in this paper

B. Algoritm 2 Implemetnation and Results

The CNN Machine Learning Model is implemented as a Sequential Model using TensorFlow meaning the model is made up of a plain stack of layers, these layers have one input and one output. The sequential model allows for adding processing layers throughout the model for tweaking performance. The first 2 Layers are 2D Convolution Layers using the Rectified Linear Unit activation function; the layers are made up of 16 and 32 nodes, respectively. Note that the Activation function of RELU was chosen due to its speed of computation enabling the team to run the models without expensive hardware. The two 2D Convolution Layers apply a sliding convolutional filter to 2-D input attempting to extract features from the 2D Array. From there the model flattens the output into a single-dimensional array. The single-dimension array supports our next set of layers. Next, we have a combination of off and dense layers. The Dense layers, known as the fully connected layers, aid in mapping the representation between the inputs and the outputs. The dropout layer protects the model from overfitting to the training data. A Summary of the Model can be found in Figure 4. [13] There is no preprocessing within Algorithm 2 as the incoming data into the AI/ML Model is already in a good state with the incoming data in Image, Head Orientation Similar, Images covered to greyscale. Note that the python script is using the ImageDataGenerator Class from the keras's preprocessing library. Meaning the gray scaled images must be separated into a hierarchal structure with the top-level splitting train and test then the next layer down grouping them into their classification.

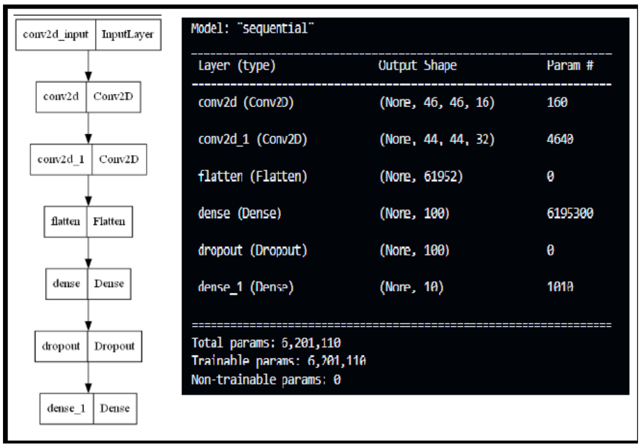

Fig. 13. Summary depticting the layers oft the Image Recoginition CNN.

After training the model using the 100 iterations of the data set you can see the accuracy of the model for test data hovers around 50% after the 15th iteration of the data set. While the validation results start to become predictable the training data results increase in value all the way through the 100 iterations. Figure 14 depicts the accuracy of the model over the iterations of the data set. Overall, it takes about 30 seconds to run each iteration of the data set meaning 30 iterations on the entire training data set costs about 50 minutes. Overall, the iteration count could be dropped by 80% with no impact to the performance of the model meaning the actual cost of the model from a timing perspective is around 30 minutes. There is a slight issue of overtraining as the performance on the test data set slowly degrades as more iterations on the training data are executed. Figure 15 below depicts the confusion matrix for the test data when run against the trained model. As shown, the model's accuracy is low, and other tweaking of the layers is needed for better performance. Overall, the basic CNN with a small number of layers was not highly successful when trying to identify emotion recognition from a static human face. Lastly figure 16 shows there is lower precision and recall on negative emotions (Fear, Sad, and Angry) as opposed to positive emotions (Happy Surprise).

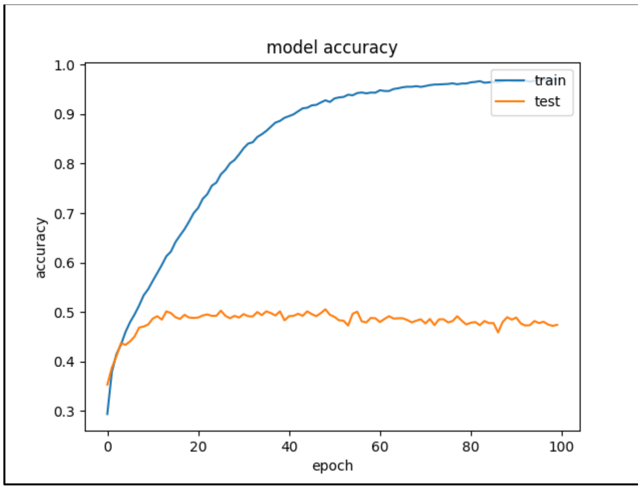

Fig. 14. Summary depticting the layers oft the Image Recoginition CNN.

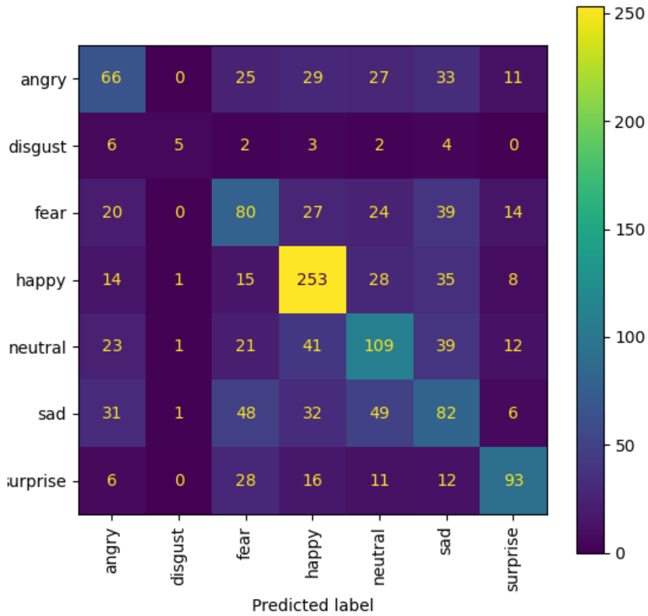

Fig. 15. Test/Validation Data Confusion Matrix for the seven classes.

	precision	recall	f1-score	support
angry	0.40	0.35	0.37	191
disgust	0.62	0.23	0.33	22
fear	0.37	0.39	0.38	204
happy	0.63	0.71	0.67	354
neutral	0.44	0.44	0.44	246
sad	0.34	0.33	0.33	249
surprise	0.65	0.56	0.60	166
accuracy			0.48	1432
macro avg	0.49	0.43	0.45	1432
weighted avg	0.48	0.48	0.48	1432

Fig. 16. Test/Validation of Classification Report for the seven classes

V. Conclusion

When comparing the SVM to the CNN, there are 3 criteria that should be considered: Accuracy, precision/recall/etc. Metrics, and Timing Cost. When dealing with Accuracy the CNN Model performed better by 7% then both types of the SVM Models. When dealing with the weighted average precision/recall on percentage in total the CNN outperformed the SVM by 7%. When looking at individual classes the SVM Radial Basis Classifier Model performed better on the Disgust emotion but the same or worse on the others. The SVM Polynomial Classifier Model performed worse on all emotions than the CNN Model. Lastly on timing, the CNN Model costs half the time for training, and a faction of a percentage for prediction execution. We have determined these results are inclusive as the CPUs/GPUs running and training the models were different, as well as the CNN was trained and executed on a local machine while the SVM Models were trained and run on a remote machine. In the end, based on the data when comparing these models, the CNN Model performed better, but added testing and model updates would be needed to come to a strong conclusion.

ACKNOWLEDGMENT *(Heading 5)*

We thank Dr. Gregory Wagner (Syracuse University) for his guidance and teaching throughout our research project. Dr. Wagner provided us with the required material and datasets necessary for conducting the experiment and research on facial recognition techniques widely published through papers and articles. We would like to thank Syracuse University for providing us with access to the necessary cloud-based technologies and research articles to implement machine learning algorithms.

REFERENCES

[1] P. Ekman, H. Ellis, D. Perrett, and E. Rolls, “Facial expressions of emotion: An old controversy and new findings,” *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 335, no. 1273, pp. 63–69, 1992.

[2] M. Grahlow, C. I. Rupp, and B. Derntl, “The impact of face masks on emotion recognition performance and perception of threat,” *PLOS ONE*, vol. 17, no. 2, 2022.

[3] S. Giri, G. Singh, B. Kumar, M. Singh, D. Vashisht, S. Sharma, and P. Jain, “Emotion detection with facial feature recognition using CNN & OpenCV,” *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2022.

[4] P. Tarnowski, M. Kołodziej, A. Majkowski, and R. J. Rak, “Emotion recognition using facial expressions,” *Procedia Computer Science*, vol. 108, pp. 1175–1184, 2017.

[5] S. Saha, “A comprehensive guide to Convolutional Neural Networks- the eli5 way,” *Medium*, 16-Nov-2022. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Accessed: 19-Feb-2023].

[6] V. Hosur and A. Desai, “Facial emotion detection using convolutional neural networks,” *2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon)*, 2022.

[7] R.amkumar, Madupu, Chiranjeevi Kothapalli. Yarra, Sonti Harika, and C. Makzeelan Basha, “Automatic human emotion recognition system using facial expressions with convolution neural network,” *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2020.

[8] S. Annadurai, M. Arock, and A. Vadivel, “Real and fake emotion detection using enhanced boosted support vector machine algorithm - multimedia tools and applications,” *SpringerLink*, 16-Jun-2022. [Online]. Available: https://link.springer.com/article/10.1007/s11042-022-13210-6. [Accessed: 20-Feb-2023].

[9] R. Kabir, Y. Watanobe, M. R. Islam, K. Naruse, and M. M. Rahman, “Unknown object detection using a one-class support vector machine for a cloud–robot system,” *MDPI*, 10-Feb-2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/4/1352. [Accessed: 20-Feb-2023].

[10] A. B. YILMAZ, Y. S. TASPINAR, and M. Koklu, “Classification of malicious Android applications using naive Bayes and support vector machine algorithms,” *International Journal of Intelligent Systems and Applications in Engineering*. [Online]. Available: https://www.ijisae.org/index.php/IJISAE/article/view/2010. [Accessed: 01-Mar-2023].

[11] A. Saxena, A. M. Alshamrani, A. F. Alrasheedi, K. A. Alnowibet, and A. W. Mohamed, “A hybrid approach based on principal component analysis for power quality event classification using support Vector Machines,” *MDPI*, 05-Aug-2022. [Online]. Available: https://www.mdpi.com/2227-7390/10/15/2780. [Accessed: 01-Mar-2023].

[12] T. Shah, “About train, validation and test sets in machine learning,” *Medium*, 10-Jul-2020. [Online]. Available: https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7. [Accessed: 20-Feb-2023].

[13] “Tensorflow dense: How to use function tensorflow dense?,” *EDUCBA*, 12-Jul-2022. [Online]. Available: https://www.educba.com/tensorflow-dense/. [Accessed: 27-Feb-2023].