Comparative Study Of Various Machine Learning Techniques To Recognise Facial Emotions.

Shivam Singhal
Computer Science Dept.
Delhi Technological University

Shashank Chugh
Computer Science Dept.
Delhi Technological University

***Abstract*— Facial emotion recognition has very wide applications in various fields like Healthcare, Security and Robotics. Even after a few decades of research, facial emotion recognition is still a very challenging problem due to high variations in images of a particular emotion. Most related research papers work extremely well on datasets in which faces are posed like CK+, but do not perform fairly well on datasets having partial, unposed low quality images like FER2013. We built several models capable of recognizing seven basic emotions (happy, sad, angry, fear, surprise, disgust, and neutral) from facial expressions which have been trained on Extended Cohn Kanade as well as the FER-2013 dataset. On CK+, our best model achieved an accuracy of 93%and in FER-2013 due to wild setting unposed images the best accuracy achieved was 66.4%.**

Keywords*— *Facial Emotion Recognition (FER), Convolutional Neural Network (CNN), k-Nearest Neighbor (k-NN), Support Vector Machine (SVM).

I. INTRODUCTION

Emotions are a crucial factor in communication between us. one. Able to recognize emotion of a person becomes important because of numerous applications in several fields, however not restricted to but not limited to, human-computer interface, animation, medicine and security. According to different surveys verbal components convey one-third of human communication, and nonverbal components convey two-thirds, also among several nonverbal components, facial expressions are one of the main information channels in interpersonal communication. FER refers to Facial Emotion Recognition.
Facial Emotion Recognition has been an active research area for a few decades but still the results are not very impressive. Major challenges faced by researchers is detecting features of the face if images are not posed which is the actual case in real life problems, another problem is very high variations in faces that depict same emotions. It is also observed in already published works that models are able to recognise a few emotions with very accurately but not able recognise other emotions with similar accuracy. We have majorly worked towards comparative study of various Machine Learning Techniques in classifying emotion of a person from happy, sad, angry, fear, surprise, disgust, and neutral by using facial expressions. As a part of preprocessing we have implemented face detection and image alignment using Haar Cascade and geometry. Aligning the face is necessary as it enhances the accuracy of machine learning models. Then we have implemented Nearest Neighbours algorithm, Support Vector Machine and Convolutional Neural Networks and later we have compared their results.

II. RELATED WORK

A lot of research has already been done in Facial Emotion Recognition already, we have referred to various papers which use various techniques like KNN, SVM and CNN.

First research paper that we studied was on Real-time Emotion Recognition from Facial Expressions by Minh-An Quinn, Grant Sivesind, Guilherme Reis from Stanford University in which CNN and SVM were trained on CK+ as well as FER2013 [4]. They have also implemented real time classification by using OpenCV's Haar cascades to detect and extract a face region from a webcam video feed, then classified it using a CNN model.
Results that we have obtained on various algorithms completely resonate with this research paper except SVM on CK+, it has been stated in the paper that by scaling the pixels and using a linear SVM accuracy on testing was close to 98% which is far more than what we were able to achieve.

Durgesh K. Srivastava and Lekha Bhambhu, Data Classification Using Support Vector Machine [6] discusses excellent work to improve accuracy of SVM classifiers. They have explained usage of kernels and tuning of various parameters to improve SVM model for classification.

Facial Expression Recognition using Deep Learning by Raghu Vamshi N and Bharathi Raja S [2]. They have used ResNet50, Transfer Learning and Ensemble Learning and trained on the FER2013 dataset and have obtained accuracy in the range 70% to 75% on the testing set, which is very good for a challenging dataset like FER2013.

FER2013 Dataset was introduced in a Kaggle competition which was part of ICML workshop in 2013 The top three teams in the competition used CNN's in combination with image transformations. The winner of the competition Y. Tang used the primal formulation of SVM as a loss function for training and also used the L2-SVM loss function [7]. This gave great results at the time, achieving an accuracy of 71.2% to top the competition.

From past related works it has been observed that proposed models are highly accurate in predicting emotions for posed datasets like CK+, but do not perform well on unposed and wild setting datasets like FER2013.

III. DATASET

We used mainly two datasets to train our models: FER-2013 and CK+ (extended Cohn-Kanade).

FER-2013:

FER-2013 Dataset was introduced in a Kaggle Competition "Challenges in Representation Learning: Facial Expression Recognition Challenge". The winner of this challenge obtained

71% accuracy. This dataset is used in various competition and research papers.

This dataset consists of over 35000 images.

This is one of the very challenging datasets with human-level accuracy of 65±5%, and the highest performing published works were able to achieve 71±2%.

Wild setting unposed images make this dataset one of the most challenging datasets. Each image in FER-2013 is labelled as one of seven emotions: happy, sad, angry, afraid, surprise, disgust, and neutral, with happy being the most prevalent emotion, providing a baseline for random guessing of 24.4%. The images in this dataset consists of posed and unposed faces. All images are in grayscale, and of 48*48 pixels. It was created with the help of Google Image Search of each emotion.

TABLE I. Distributions of Images in the FER-2013 dataset.

Label	Number of Images	Label	Number of Images
Anger	4953	**Sad**	6077
Disgust	547	**Surprise**	4002
Fear	5121	**Neutral**	6198
Happy	8989	**Total**	35887

Fig. 1. Sample Images from each class of the FER2013 Dataset.

CK+:

The extended Cohn-Kanade (known as CK+) facial expression database is a public dataset for emotion recognition. The CK+ comprises a total of 981 images in total, the majority of them are posed frontal face images. The images are this dataset too in grayscale, and of 48*48 pixels. All of them are frontal posed images.

TABLE II. Distributions of Images in the CK+ dataset.

Label	Number of Images	Label	Number of Images
Anger	135	**Happy**	207
Contempt	54	**Sad**	84
Disgust	177	**Surprise**	249
Fear	75	**Total**	981

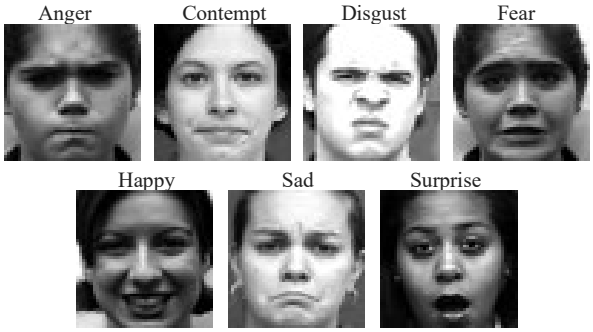


Fig. 2. Sample Images from each class of the CK+ Dataset.

IV. EXPERIMENTAL DESIGN

A. Holdout Validation Technique -

We split the FER-2013 dataset into a training set of 29068 images, a validation set of 3230 images and a testing set of 3589 images.
The CK+ dataset splits into a training set (784 images) and a testing set (197 images) only due to a smaller number of images in the dataset.
We ran all our models in Google Collaboratory storing the dataset and other necessary files in Google drive.

B. Performance Metrics -

Area under curve (AUC), Precision, Recall gives better intuition of results as compared to accuracy. We can calculate Precision, Recall, F1-score by the below given formulas once Confusion Matrix is formed.
The evaluation metrics for FER and CK+ Dataset are classified into four methods using different attributes:

a) Accuracy is defined as the ratio of true prediction to the total number of images in the set.

$$acc = (TP + TN)/Total\ Population$$

b) Precision is the fraction of predicted emotions that are correctly recognized.
Precision is defined as –

$$P = TP/(TP + FP)$$

c) Recall simply is the fraction of no. of correct prediction of a particular emotion over the actual no. of images with that emotion. Recall is defined as –

$$R = TP/(TP + FN)$$

d) F1-score has predictive power in terms of spatial consistency.
F1-score is defined as –

$$f1 - score = (2 \times R \times P)/(R + P)$$

TP is no. of true positives i.e. number of correct classified emotions, FN is no. of false negatives, and FP is the no. of false positives.

C. Preprocessing

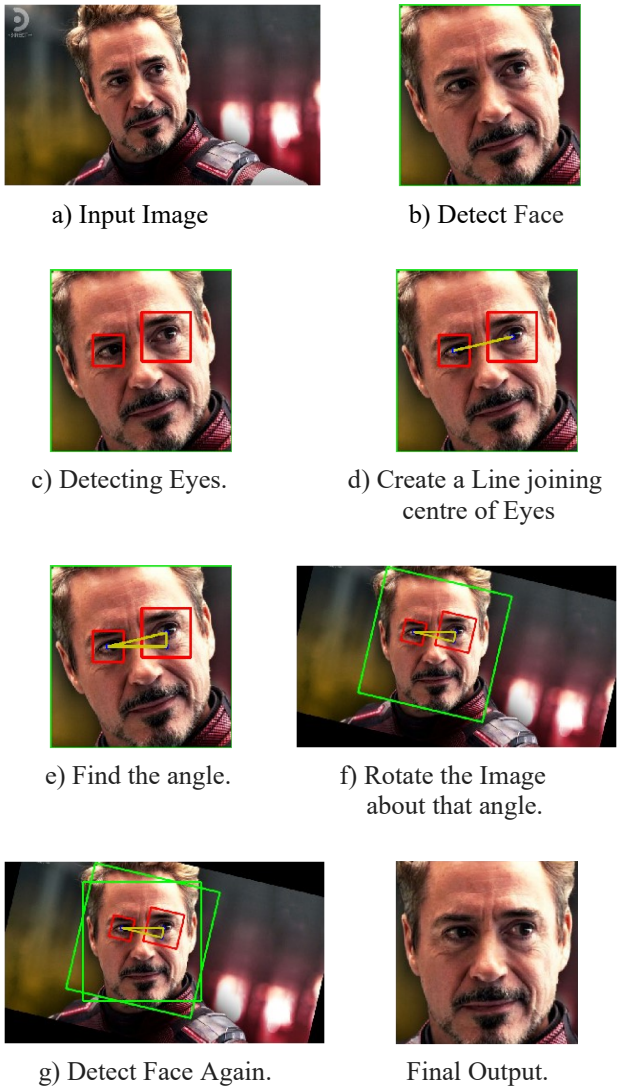
Face Alignment –

Some facial features like eyes, lips are the most important features in determining the emotion of a person, but it was observed that both CK+ and FER2013 have many images in which faces are not aligned horizontally. These unaligned

images can affect our model's accuracy, so to tackle this problem we align Faces in our dataset using OpenCV and Geometry.

Procedure (as shown in Fig.3) –

a. Do iteration for all images in the dataset.
b. Detect face and crop the image accordingly.
c. Detect eyes in the cropped image.
d. Create a line that connects the centre of two eyes.
e. Find the slope of this line with respect to the horizontal axis.
f. Rotate this image by the angle obtained in the previous step, to get an aligned image.
g. Detect Face again and crop in the aligned image.



a) Input Image
b) Detect Face
c) Detecting Eyes.
d) Create a Line joining centre of Eyes
e) Find the angle.
f) Rotate the Image about that angle.
g) Detect Face Again.
Final Output.

Fig. 3. Procedure of Face Alignment.

The mathematical formula to find the angle (shown in Fig. 4) –

Coordinates of Left Eye - $x_l,\ y_l$
Coordinates of Right Eye - $x_r,\ y_r$

$$\Delta x = x_r - x_l$$
$$\Delta y = y_r - y_l$$
$$\tan\theta = \frac{\Delta y}{\Delta x} \implies \theta = \tan^{-1}\left(\frac{\Delta y}{\Delta x}\right)$$

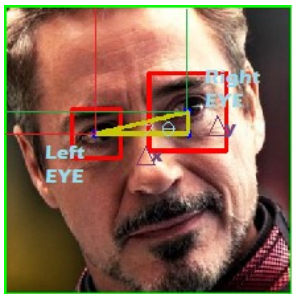


Fig. 4. Derivation of angle formed by the line joining centre of Eyes.

D. ***Algorithms -***

k-Nearest Neighbours:

k-NN uses all the training data for classification. As it iterates completely on training examples every time it wants to predict a testing example. Thus, it takes more time than other algorithms. Basically, in this algorithm, every image in the testing set is matched with all the images in the training set and is categorized the same as the most frequent class in the k nearest neighbours.

First, we used k-NN with k=7, but the accuracy achieved was close to 37% on FER2013 which is only 13% more than the baseline accuracy, to improve this we tried varying the value of k from 1 to 10 and the best results were close to around 41% which is a decent rise, but still, accuracy is very less so we tried another variant of k-NN, which is weighted k-NN, highest accuracy achieved was around 43% which is still very less (shown in Fig. 5 and 6).

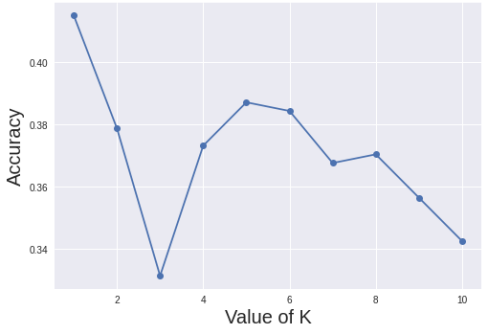


Fig. 5 Accuracy vs Value of K in k-NN

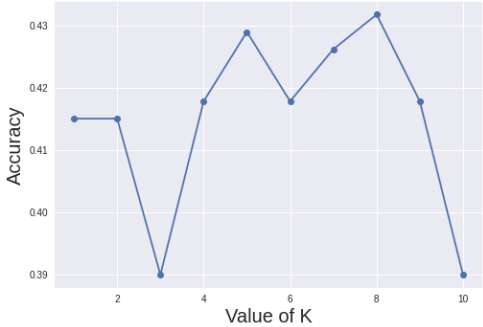


Fig. 6 Accuracy vs Value of K in Weighted k-NN

Support Vector Machine:

Support vector machine is a classification algorithm developed by Vapnik at Bell laboratories, which was primarily used for binary classification, but the idea can be extended for multi classification. The core purpose of SVM is to separate the data with decision boundary and extend it to non-linear boundaries using kernel trick. By using kernel trick SVM is able to separate non-linearly separable data by generating a hyperplane
and two more planes parallel to it on each side of the hyperplane such that the margin between these planes is maximum.
One of the major challenges is that of choosing a suitable kernel for a given application, firstly we tried using linear kernel and resulting accuracy was 63%, but we came across a research paper [6] that argues benefits of rbf kernel over other kernels.

We were able to find following arguments in favour of RBF -

- RBF kernel helps to separate non linearly separable datasets, by introducing a new dimension.
- RBF kernel has less parameters than the polynomial kernel, so it has less computation time.

By using rbf kernel and default values for regularization and gamma parameter, the result obtained was close to 67%.
We wanted to implement parameter tuning, but due to large number of features, computation time was very large so before using parameter tuning, we used Principal Component Analysis to reduce the number of features in the dataset. 95% variation was conserved and resulting features in the dataset were 101 from initial count of 2404, which drastically improved the computation time for SVM and thus helped us to experiment more in parameter tuning.
Then we used parameter tuning for regularization and gamma parameters using Grid searching technique and 5-cross validation, resulting in 70% accuracy on testing data.
It was realized that, if we could use a larger range for the parameters, accuracy might increase. But increasing the count of possible values for various parameters will result in large computation time, so instead of Grid Search we used Randomized Search with a larger set of parameters and 3000 iterations and 5 cross validation. As expected, accuracy improved to 75%.

In FER2013, the best accuracy obtained was 46% using rbf kernel and default values of parameters. Due to the very large size of the dataset we were not able to perform parameter tuning.

Convolution Neural Network:

CNN is a deep learning method that is broadly used for image classification, image recognition, object detection, etc. Our CNN Model contains three types of heterogeneous layers: convolution layer, max pooling layer, and fully connected layers. Conv. Layer takes image vector as input and convolves these inputs using some filter in a sliding window manner to give output matrix. Max Pooling Layers lowers resolution of image vector by max-pooling the given input vector to reduce their dimensions and ignore variations in geometric distortions. Fully connected layers compute scores of the entire image,

The first model we built consisted of two 3x3x64 same padding convolution layers, two 3x3x128 convolution layers with RELU activations, two 2x2Maxpooling layers, and completed with an FC layer and a SoftMax layer. Also, batch normalization and 25% dropout layers were included to achieve an accuracy of 55%, to estimate the potential of CNN, we used VGG16 architecture, VGG16 is deep model CNN architecture proposed by Karen Simonyan and Andrew Zisserman, Oxford University in 2014 in the paper "VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE SCALE IMAGE RECOGNITION" [1].

VGG 16 is a deep model architecture that consists of 16 weight layers including thirteen convolutional layers with a filter size of 3 X 3, and fully-connected layers. Due to a large number of layers, our model started overfitting the dataset very early, so we removed five convolutional layers from the architecture. Now our architecture consists of 8 conv. Layer of filter size 3 X 3, four Max Pooling Layers, and 2 Fully connected layers with SoftMax layer as output (shown in Fig. 7).

After that validation accuracy after 30 epochs was close to 66%.

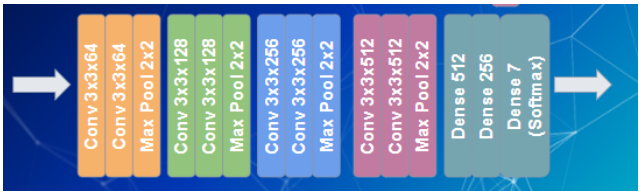


Fig.7. Architecture of our Model.

V. RESULTS AND ANALYSIS

As discussed earlier also, the FER-2013 dataset is much more challenging than CK+ due to Intra-class variation of FER, another challenge of this dataset is the imbalance nature of various categories, a number of categories like happiness and neutral have plenty additional examples than others.

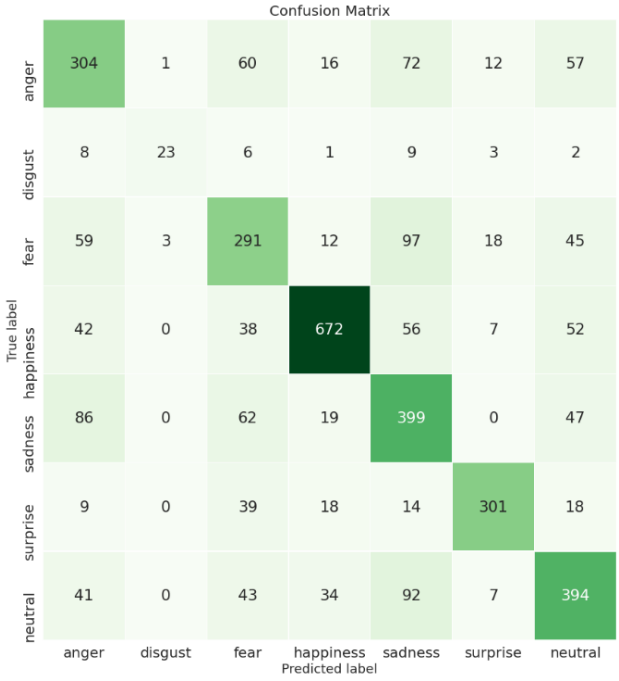


Fig. 8: The confusion matrix of the CNN model on the test set of FER-2013 Dataset.

*Generated with visualization code from sklearn.

We used the entire 29,068 images in the training set to train the model, validated on 3230 validation images, and reported the model accuracy on the 3,589 images in the test set. We were able to achieve an accuracy rate of around 66.4% on the test set.
The confusion matrix is shown in Fig. 8. The confusion matrix is obtained on the test set of FER-2013 Dataset.

With the help of the Confusion Matrix we can see that the model is making mistakes in anger and sadness. With a true label as anger, the model is predicting 72 examples as sadness and vice versa 86 sadness examples as anger. Also, disgust, anger and fear examples are very less compared to others. Thus, our model makes more mistakes in those classes. Our model shows best results on happiness. This is the most reliable classification.

A. CONFUSION MATRIX

On the CK+ It is observed from Fig.9, that our model shows good results on almost every emotion. Hence this validates the high accuracy achieved on CK+.

On the CK+ dataset, with a small number of posed, centred photos, we achieved a test accuracy of 95.4%. Our model is much accurate and doesn't confuse between classes, but on FER2013 due to wild setting unposed images results are not that good and accuracy achieved is 66.4%.

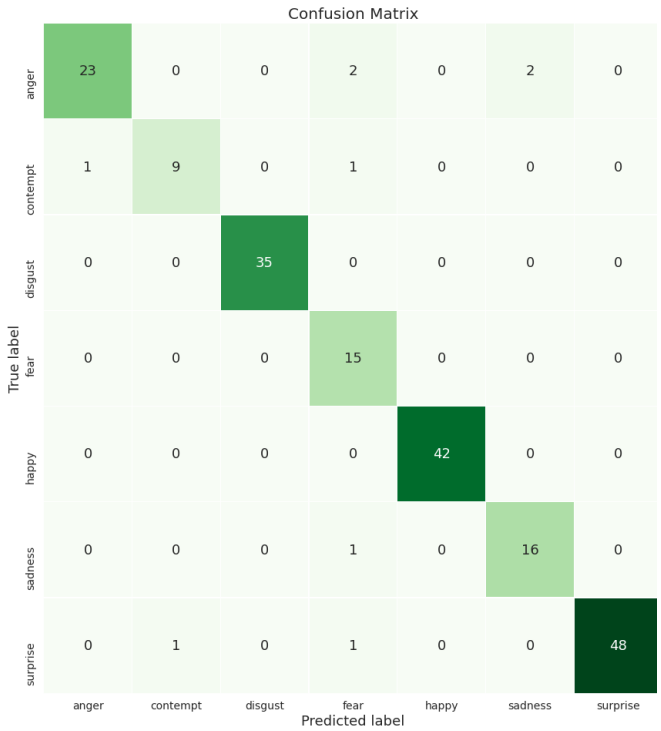


Fig. 9: The confusion matrix of the CNN model on the test set of CK+ dataset.

B. MODEL ACCURACY AND LOSS OVER TRAINING PERIOD

As we see from the fig. 10, there is a flattening of accuracy at 30 epochs. Due to early stopping our model stops after 30 epochs as validation loss is not decreasing further. After that there will be overfitting of data.

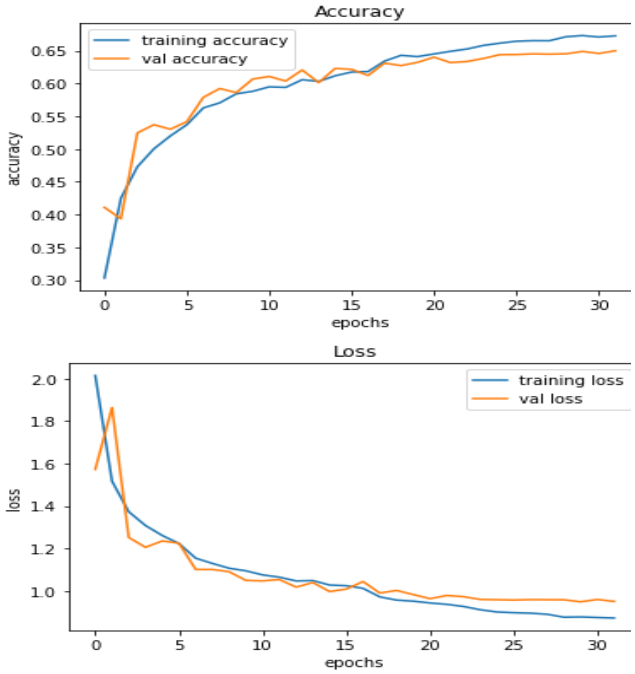


Fig. 10: Model Accuracy and Loss over training period of FER 2013 Dataset

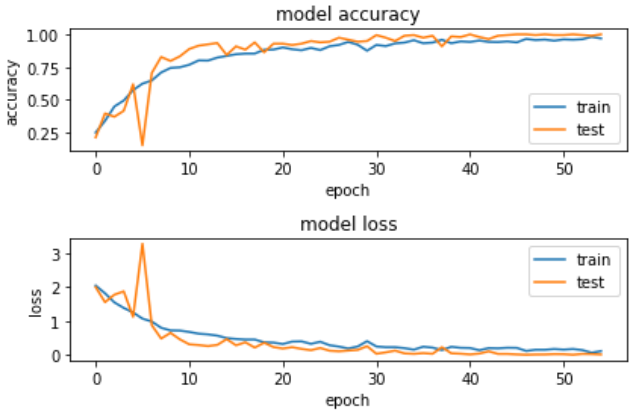


Fig. 11: Model Accuracy and Loss over training period of CK+ Dataset.

TABLE III. – Classification accuracies, precision, recall and f1-score for FER2013 and CK+ test dataset on our CNN model.

Dataset	Accuracy	Precision	Recall	F1-score
FER 2013	66.4%	69.4%	62.9%	65.99%
CK+	95.4%	93%	94%	93%

On CK+ dataset AUC score calculated was around 99% which is a very good results in comparison to other works.

VI. CONCLUSION AND FUTURE WORK

The main objective of this paper was to compare various machine learning techniques in classifying emotion of a person. We were able to cross baseline accuracy 24.4% by k-NN only but to surpass human level accuracy we implemented some advanced techniques. So, we tried SVM and CNN. On the FER-2013 dataset, we achieved a test accuracy of 66.4% with our best

CNN, which is decent in comparison with the highest accuracy achieved on FER2013, that is 71%. On CK+ results were much better due to posed images. Our system reliably classified some emotions (the most reliable classification being happy), but didn't perform well on other emotions ,this indicates that one challenge that real time emotion recognition faces is that the features of different emotions (like disgust, fear, neutral) are very similar, and more analysis on ways to separate similar emotions may be needed to improve the model**.** Our model also struggled when lighting conditions were changed, or backgrounds were noisy. Though we train on FER-2013 to better reflect real-time data. In conclusion, to implement FER in some real-world problems we need a better model which performs fairly well on unposed images as well.

VII. REFERENCES

[1] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[2] Raghu Vamshi N and Bharathi Raja S. Facial Expression Recognition using Deep Learning. IEEE

[3] Shervin Minaee, Amirali Abdolrashidi, Expedia Group University of California, Riverside. Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network. arXiv:1902.01019v1 [cs.CV] 4 Feb 2019.

[4] Minh-An Quinn, Grant Sivesind, Guilherme Reis CS 229 - Stanford University. Real-time Emotion Recognition from Facial Expressions.

[5] Byoung Chul Ko ID Department of Computer Engineering, Keimyung University, Daegu 42601, Korea. A Brief Review of Facial Emotion Recognition Based on Visual Information. MDPI 2018.

[6] Durgesh K. Srivastava and Lekha Bhambhu, Data Classification Using Support Vector Machine