# Face Recognition And Facial Expression Recognition

Het Jagani, 1641007, Hitarth Panchal, 1641034,
Malak Khandwala, 1641031, Maulik Swaminarayan, 1844006

***Abstract*—Face Recognition problem has been around since a long time. We are using Machine Learning techniques for better performance and accuracy. Data augmentation and pre-processing of the dataset is done, to increase the size and diversity of dataset. Face Recognition begins with extracting the features such as width of mouth, width of eyes, pupil, and compare the result with the measurements stored in the database and return the closest record (facial metrics). This extraction of face coordinates can be done through many methods such as SVD. In this project a pre-trained deep neural network is used for this purpose. After the extraction of the features various classification techniques are implemented to find the match between two images for Face Recognition. In this project various classifier algorithms like ANN, SVM and KNN are experimented and evaluated for the classification task. For emotion recognition task has different problems. To achieve more accuracy the dataset for different emotions must be diverse and should be large enough so that the emotion features can be trained. For this problem Convolutional Neural Network algorithm is used, so the algorithm does feature extraction in the CNN layers. Experimentation of different network architectures and hyperparameters is done to improve the accuracy of the model.**

**keywords : Convolutional Neural Network (CNN); Artificial Neural Network (ANN); Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Transfer Learning, Fine-tuning, Data Augmentation.**

## I. Introduction

Developing a system of face recognition is quite difficult, because faces are complex, multidimensional, which contains meaningful visual features of an individual. Reason behind the complexity is the quality of the photo, angle at which its taken, illumination in the photo and many time varying changes to the faces of an individual. These all things are very important while designing such a system [1]. In the worst conditions there are high chances of failure, especially when we are doing it on large set of classes.

Due to lack of data and computing resources, to achieve better accuracy in Face Recognition problem, was incredulous. But due to Machine Learning techniques and modern Neural Network algorithms, the accuracy can be achieved. Face recognition can be used in many areas such as for filtering, bio metric fingerprinting (as everyone has unique face), automatic video surveillance. Each tasks has their own set of challenges, but the basic problem is to achieve high accuracy in face recognition task. This can be achieved using modern Machine Learning algorithms.

Face recognition task can be broken down into following parts [2]:

- Data Augmentation.
- Facial Landmarks Detection.
- Alignment and Extraction of the face.
- Extraction of facial coordinates.
- Face Classification

## II. Literature Survey

When it comes to do face recognition in real time many approaches fail. [2] represents an efficient way using to do face recognition in real time using modern deep learning. Which involves doing many preprocessing steps on an image before feeding the image to a deep neural network. State of the art face recognition techniques requires many preprocessing steps to be performed on the dataset. First step is face detection in the image. [3] uses dlib python library to do a face detection and as a result it gives the bounding box coordinates in the image. Though this approaches sometimes fail badly in bad lighting conditions. [4] shows an deep learning approach to detect faces in image. It uses pretrained model made in caffe to do all the work. Another problem is absence of large dataset. For an model to perform good on diverse images, each class should contain thousands of images taken at different places, in different lighting conditions and in different poses. [5] shows some data augmentation techniques which can augment the dataset significantly. After all it all comes to deep learning model. [6] and [7] uses diffrent kind of convolutional neural network architectures to do facial expression recognition.

## III. Data Pre-processing

### A. Data Augmentation

The dataset in this project is collected from the members in the class. For that each person in the class has provided 7 different images having different emotion (Neutral, Happy, Sad, Surprised, Anger, Fear, Disgust). There are total 62 people in the class hence the size of our dataset is $62 \times 7 = 434$ images. The dataset contained RGB and Sketch images, but we have considered only RGB images for out problem. The size of the dataset was too small, therefore various data augmentation techniques are applied.

Following are the transformations that are applied on each image for data augmentation purpose [5] Data augmentation factor is written in bracket:

- Flipped image right left ($\times 2$)

- Rotated images $30^o$ right and left ($\times 3$)
- Added Gaussian Noise ($\times 2$)

After these augmentation techniques, the size of the dataset is approximately 5000.

### B. Facial Landmarks Detection and Alignment

This step is done to align all the faces in each image. For that, first step is to detect different landmarks of the face such as position of eyes, nose, eyebrows, mouth, jawline etc. All these detected landmarks are aligned to the center of frame for all the faces. Through this step we are able to extract out the face in center of frame and all its unique landmarks.

For detection of landmarks and alignment of face we have used a Deep Neural Networks model implemented in Caffee. Dlib library is also used to implement the same. Then we chose the best of both images for further process. In this process we give our input image of any size and the model generates output image of size $96 \times 96$ with the face aligned in center of the frame.

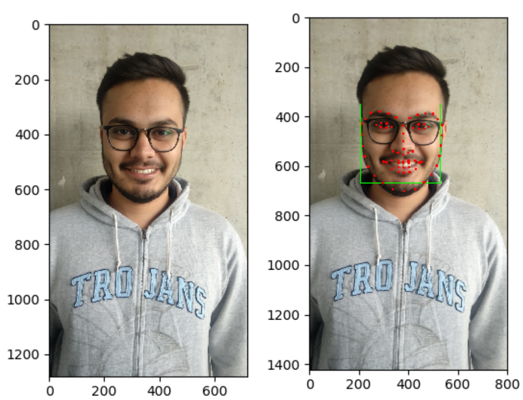

Fig. 1. Left: Input image, Right: Facial Landmarks and bounding box detected in Image.

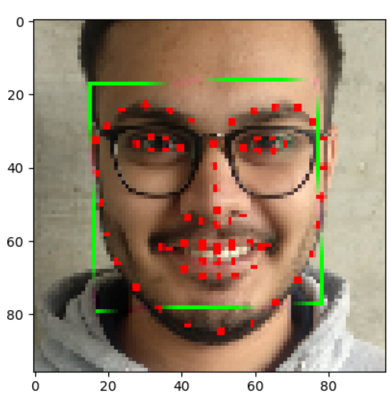

Fig. 2. Facial Landmarks aligned to the center of frame ($96 \times 96$ image).

### C. Extracting Face Embedding

To represent the unique features of each and every different faces some kind of encoding of face needs to be done. For this various techniques such as Principle Component Analysis(PCA) using SVD is used. This way we can achieve dimensionality reduction for further use in the classifier. In this project this task is achieved using a pre-trained deep neural network model [8]. This model is trained on much larger dataset of faces. The model takes $96 \times 96$ RGB image and converts it to 128 dimensional feature vector. This pre-trained model is used because it gives us more precise and unique features of the face.

## IV. Face Classification

After getting the encodings of the face, now we are required to classify the images based on classes. The classes are defined based on the problem that we are dealing. For eg. if we are doing gender classification then our classes would be binary (i.e. Male and Female). Here, as the problem is of face recognition our classes would depend on the number of different people in our dataset. In our case, the dataset consist of 62 different persons.

For face recognition, many different algorithms can be used on the encodings that are obtained from our dataset. For eg, for an input image we can find euclidian distance of that image's encodings with all the other images and based on minimum value we can find the match. But this kind of algorithm would perform very poorly for normal applications. Hence, we are using different machine learning algorithms for the same task. Following are the classifiers that we have experimented.

### A. Artificial Neural Network

We have implemented a neural network that takes the 128 dimensional encoding vectors as input and outputs the 62 dimensional vector with probabilities for each class. Following is the architecture of the model that we used.

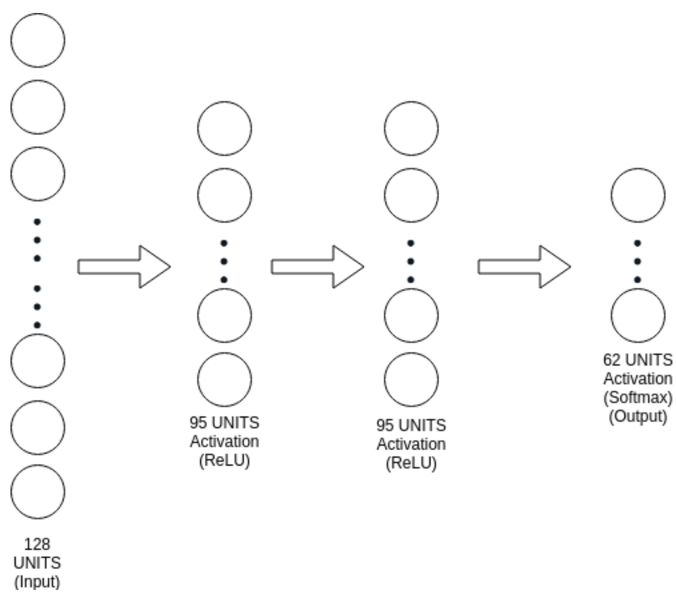


Fig. 3. Architecture of the ANN model.

For training the dataset, to avoid overfitting of the data we have implemented dropout regularization with keepprob=90%

at each layer. Also before providing input vectors, mean normalization is applied on the whole dataset. After training the classifier using Adam optimizer with batch size 16 and for 100 epochs, we are getting 98% accuracy on training dataset and 91% accuracy on test dataset.

### *B. Support Vector Machine*

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane [9]. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side (Figure 4). SVM method is based on the following idea: The input vectors are mapped into a high-dimension space Z via certain non-linear transformation, in which the optimal hyperlane can be constructed.

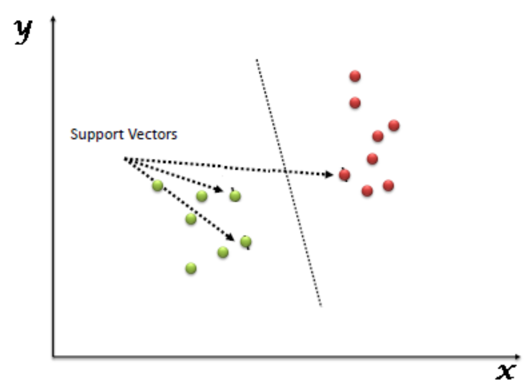


Fig. 4. Support vector machine.

**Radial Basis Function :**

$$K(x, x') = exp\Big(-\frac{||x - x'||^2}{2\sigma^2}\Big) \tag{1}$$

By training SVM based model with C parameter values 1000, 5000, 10000, 50000 and $\gamma$ values 0.0001, 0.001, 0.01, 0.1, we have chosen the best values. The kernal used is Radial Basis Function(RBF). We are getting best results for C = 1000 and $\gamma$ = 0.01. Accuracy on test dataset is 96% and accuracy on training dataset is 99%.

### *C. K Nearest Neighbour*

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions) [10]. In this algorithm we have to give K as parameter, which is the number of neighbours with which the algorithm will calculate the similarity. And based on the similarity measures with different neighbours the class of particular example is decided.

In this project we have implemented KNN classifier with K parameter 3. We have considered similarity measure as euclidian distance. After training the model. We are getting 90% accuracy on test dataset and 99% accuracy on train dataset.

## V. Facial Expression Recognition

Humans interact with each other mainly through speech, but also through body gestures, to emphasize certain parts of their speech and to display emotions. One of the important ways humans display emotions is through facial expressions which are a very important part of communication. Though nothing is said verbally, there is much to be understood about the messages we send and receive through the use of nonverbal communication. Facial expressions convey non-verbal cues, and they play an important role in interpersonal relations. Automatic recognition of facial expressions can be an important component of natural human-machine interfaces; it may also be used in behavioral science and in clinical practice.

### *A. Implementation Method*

Data pre-processing and Data augmentation steps are same as before for the facial emotion classification problem. We have used VGG19 pre-trained model for emotion classification task, but due to very much complexity of the model, less diversity of data set, and less examples in our dataset, the model was not converging. Second we used the alex-net model architecture. We implemented the model using keras library and trained it on our dataset. But we faced the same problem as before. The model parameters were not converging.

For this problem we have used the Convolutional Neural Network model [7]. The input of the model is 48 $\times$ 48 grayscale image. So after data preprocessing, we have further resized the image to 48 $\times$ 48 and converted it to grayscale image.

We trained Deep Convolutional Neural Network with 4 convolutional layers and two Fully Connected layers. The first convolutional layer had 64 3$\times$3 filters, the second one had 128 5$\times$5 filters, the third one had 512 3$\times$3 filters and the last one had 512 3$\times$3 filters. In all the convolutional layers, we have a stride of size 1, batch normalization, dropout, max-pooling and ReLU as the activation function. The hidden layer in the first Fully Connected layer had 256 neurons and the second Fully Connected layer had 512 neurons. In both Fully Connected layers, same as in the convolutional layers, we used batch normalization, dropout and ReLU. We used Softmax as our activation in output layer. we used categorical cross entropy as loss function. We have divided our dataset to training and testing, 80% and 20% respectively. Figure 5 shows the architecture of the deep model. To train the model we have used adam optimizer.

## VI. Results

We have tested different classification algorithms for the face recognition problem. Due to less number of examples in our dataset, we are able to achieve good accuracy only for the given dataset examples. We have experimented three

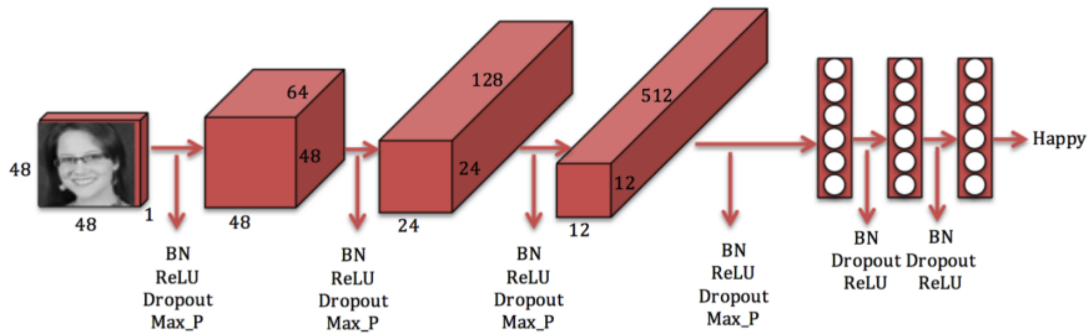


Fig. 5. Architecture Deep CNN model

different classification algorithms after feature extraction. By using K Nearest Neighbour we are getting 90% accuracy on the test dataset, by using SVM classifier we are getting 96% accuracy on the test dataset, and by using Neural Network algorithm we are getting 91% accuracy on the test dataset.

For the emotion detection task, we have experimented with different model architectures and hyper-parameters to achieve good accuracy. As there is less diversity in the dataset, we are able to get 85% accuracy on the test dataset after training the model for 60 epochs. But this accuracy is only limited for our given dataset, because the size of given dataset is small. Also the dataset contains the images for which the model is getting confused. For eg. some of the fear face is same as the surprise or neutral face. This correlation between the emotions can be seen from the confusion matrix as shown in Figure 6.

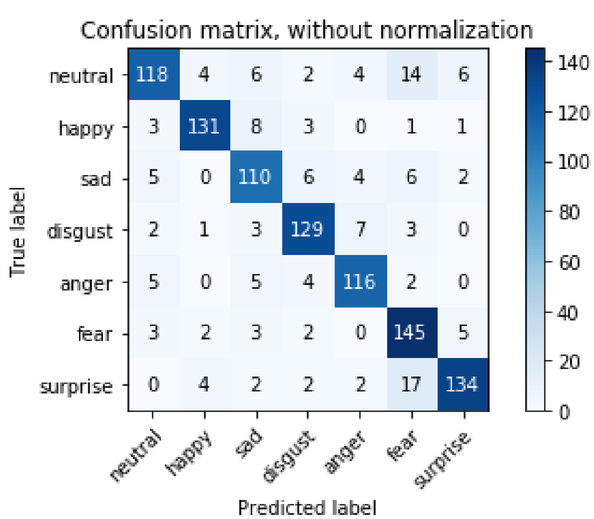


Fig. 6. Confusion Matrix for emotion recognition.

## VII. Conclusion

Problem of face recognition has been approached by many people in many ways, but always the problem of over-fitting remains. Hence the accuracy achieved in such task is limited. In this project, we have achieved high accuracy using clustering algorithms such as SVM. But the accuracy is high only for the given dataset. This is due to less number of examples in the dataset and also less diversity in the dataset. For better performance we should have large dataset with 1000-1500 images of single person with different alignment and pose.

We developed various CNNs for a facial expression recognition problem and evaluated their performances using different post-processing and visualization tech- niques.The results demonstrated that deep CNNs are capable of learning facial characteristics and improv- ing facial emotion detection. Also, the hybrid feature sets did not help in improving the model accuracy, which means that the convolutional networks can intrinsically learn the key facial features by using only raw pixel data.

## References

[1] A. Rosebrock, *Face recognition with opencv, python, and deep learning*, Feb. 2019. [Online]. Available: https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/.

[2] A. Geitgey and A. Geitgey, *Machine learning is fun! part 4: Modern face recognition with deep learning*, Jul. 2016. [Online]. Available: https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78.

[3] A. Rosebrock, *Facial landmarks with dlib, opencv, and python*, Nov. 2018. [Online]. Available: https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/.

[4] ——, *Face detection with opencv and deep learning*, Feb. 2019. [Online]. Available: https://www.pyimagesearch.com/2018/02/26/face-detection-with-opencv-and-deep-learning/.

[5] B. Raj and B. Raj, *Data augmentation — how to use deep learning when you have limited data - part 2*, Apr. 2018. [Online]. Available: https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced.

[6] G. S. Dan Duncan and C. English, *Facial emotion recognition in real time*, Mar. 2016. [Online]. Available: http : / / cs231n . stanford . edu / reports / 2016 / pdfs / 022_Report.pdf.

[7] S. Alizadeh and A. Fazel, "Convolutional neural networks for facial expression recognition," *CoRR*, vol. abs/1704.06756, 2017. arXiv: 1704 . 06756. [Online]. Available: http://arxiv.org/abs/1704.06756.

[8] M. Krasser, *Face recognition*, https : / / github . com / krasserm/face-recognition, 2018.

[9] J. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, Jun. 1999, ISSN: 1573-773X. DOI: 10.1023/A:1018628609742. [Online]. Available: https://doi.org/10.1023/A:1018628609742.

[10] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 4, pp. 580–585, Jul. 1985, ISSN: 0018-9472. DOI: 10.1109/TSMC.1985.6313426.