FACIAL EXPRESSION RECOGNITION: FROM NAMED EXPRESSIONS TO UNNAMED EXPRESSIONS

A Thesis

by

SAHUL MADANAYAKANAHALLI PHANIRAJ VENKATESH

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Anxiao (Andrew) Jiang
Committee Members,	Jiang Hu
	Weiping Shi
Head of Department,	Miroslav M. Begovic

August 2019

Major Subject: Computer Engineering

ABSTRACT

Facial expressions plan a very important role in interpersonal relations as they convey non-verbal cues. Automatic recognition of facial expressions forms a crucial component in human-machine interfaces. The main motivation behind choosing this problem is that there are many facial expressions to recognize. It is a very difficult task to categorize them as they are very subtle. Same expressions can have different meanings for different people in different context. Facial Expression Recognition (FER) has applications across many domains like business, education, and health care. In the existing work, people mainly focus on recognizing the seven basic expressions like happy, sad, disgust, angry, surprise, neutral, and fear. In this research work, we try to explore a new direction where we try to recognize many more expressions apart from the basic seven expressions. These expressions are hard to name but exist in real life.

The approach taken is One-shot learning. Every time we observe a new expression, we use one-shot learning technique to recall previous cases where same expression was seen. By doing this, people can understand in which context the same expression appears, which will lead to the understanding of each expression.

In the present work, we train the neural network for the basic seven expressions. We later extract the features from the penultimate Fully Connected (FC) layer as a feature representation for the input image. These features are used in further processing and as a basis for one-shot learning. While the current research involves 2D static images, we further extend our research from 2D expression to 3D video clips. The main reason for doing this is, expression is not a static image of the face at a given time. Actually, if we involve the change in expression in a short period of time, it is more meaningful in recognizing expression. This aspect has been less explored before.

The results obtained for 2D static images show that One-shot learning performs a very good job in recognizing new expressions with just one training example.

DEDICATION

To my parents

ACKNOWLEDGMENTS

I would like to express my gratitude to my thesis advisor, Professor Anxiao (Andrew) Jiang, for his continuous guidance and support. He has always been a constant motivator throughout my thesis work. I would like to thank my committee members, Professor Weiping Shi and Professor Jiang Hu for serving the comittee.

I would like to thank my friend Satish Pasumarthi for his valuable insight. In addition, I would like to thank Sachin Puranik and Ram for their support with developing GUI and sharing their knowledge on computer vision and deep learning.

I am grateful to all my family members and friends for their constant support and motivation.

CONTRIBUTORS AND FUNDING SOURCES

Contributors and Funding Sources

This work was supervised by a thesis committee consisting of Professor Anxiao (Andrew) Jiang, Professor Weiping Shi, and Professor Jiang Hu of the Department of Electrical and Computer Engineering.

This work is carried out without outside financial support.

NOMENCLATURE

FER	Facial Expression Recognition
FC	Fully Connected
DNN	Deep Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
CS	Cosine Similarity
ED	Euclidean Distance
RN	Relation Network
OSVOS	One-Shot Video Object Segmentation
LOSO	Leave One Subject Out
WCSS	Within-Cluster Sum of Square

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

In the field of computer science, vast possibilities of applications have made facial emotion recognition challenging and unavoidable. The use of non-verbal cues like facial expressions, body movement, and gestures convey the feedback and emotion to the user. The information conveyed to the other person using these non-verbal cues are always better than just speaking. Automatic recognition of facial expressions play an important role in natural human-machine interfaces . Although humans recognize facial expressions without much effort, reliable expression recognition by machine is still a challenge [2].

In the domain of computer vision and machine learning, various facial expression recognition (FER) systems have been explored to extract expression information from faces. In the early twentieth century, Ekman and Friesen [3] have defined six basic emotions based on cross-culture study. These expressions include anger, disgust, fear, happiness, sadness, and surprise [4]. The recent transition of FER from laboratory-controlled environment to challenging in-the-wild conditions along with the recent success of deep learning techniques, deep neural network have been increasingly leveraged to learn discriminative representations for the FER problem [4].

There are numerous datasets available publicly for facial expression recognition (FER) problem. In this research work, for training the neural network, we have chosen is FER2013 dataset. The motivation behind choosing this dataset is that it consists of large-scale images of facial expressions in the wild. The dataset comprises of seven basic expressions viz. happy, sad, disgust, surprise, angry, neutral, and fear [5]. The authors of [6] achieve the state-of-the-art performance over this dataset. They achieve test accuracy of 75.2%. Their model is based on convolutional neural network (CNN) and they implement network ensemble approach.

However, in real life scenario, there are more subtle expressions than the seven basic expressions that are much hard to detect. In one of the recent discoveries by scientists, it was shown that humans execute around 21 different expressions [7]. Some of these depict compound emotions like ‘angrily surprised’, ‘sadly surprised’. Apart from these compound expressions that are very exaggerated, there are many subtle expressions in real life which are mostly unnamed. It is not very clear on how to name these subtle expressions and not much work has been done in this direction. In our research, we focus on this part. In this research work, we want move from recognizing the named expressions which are limited to understanding and recognition of these unnamed expressions.

The challenge we face here is that, since the expressions are unnamed, we do not have data with labels. The approach taken here is One-shot learning. Every time we observe a new expression, we use one-shot learning technique to recall previous cases where same expression was seen. By doing this, people can understand in which context the same expression appears, which will lead to the understanding of each expression.

The road-map for the research is laid out in the following three stages. First, we develop a neural network that does the recognition of the basic seven expressions. Second, we extract the features from the penultimate full connected (FC) layer as a feature representation of the input image in the feature space. These features are used in further processing and as a basis for One-shot learning. Lastly, we extend our research from 2D static images to 3D short video clips.

2. RELATED WORK

The transition of facial expression recognition (FER) problem from laboratory-controlled facial expressions to challenging in-the-wild conditions along with the recent success in deep neural network (DNN) and deep learning techniques have been very useful in learning the discriminative features for the facial expression recognition (FER) problem. The recent FER systems based on DNN focuses on two important issues namely, overfitting caused by lack of training data and expression-unrelated variations like illumination, identity bias, and head pose [4].

In this section, we provide a brief summary on facial expression recognition (FER) systems based on deep neural network (DNN), including datasets and algorithms that provide brief insights into these problems. First, we briefly summarize the various publicly available datasets for FER problem. Second, we review the state-of-the-art DNN and related training strategies that are designed for FER problem based on dynamic and static images. Third, we summarize the existing approaches to One-shot learning. We then extend our discussion to the advantages and limitations involved in each technique. Finally, we conclude the section to explain as to why the current research work is novel and how One-shot learning helps to overcome the problem of lack of training examples.

2.1 Facial Expression Datasets

One of the critical aspect involved in training deep neural network (DNN) is to have a robust dataset. It is of utmost importance to have datsets that have sufficient labeled training data that constitute as many variations of the environments and populations as possible. In this section, we briefly summarize the publicly available datasets that contain the basic expressions and that have been used widely in the papers reviewed for evaluating deep learning algorithms [4].

CK+: CK+ dataset contains 593 video sequences from 123 subjects. The sequences vary in duration from 10 to 60 frames. They show a shift from a neutral facial expression to the peak expression in each video sequence. Among these video sequences, 327 sequences from 118 subjects are labeled with seven basic expression labels based on the Facial Action Coding System (FACS) [4].

MMI: MMI database consists of videos developed in a laboratory-controlled environment. It comprised of 326 video sequences from 32 subjects. Among these video sequences, 213 sequences are labeled with six basic expressions (without "contempt"), and 205 sequences are captured in frontal view. Additionally, MMI has more challenging conditions, i.e., there are large inter-personal variations because subjects perform the same expression non-uniformly. Additionally, many of them wear accessories (e.g., mustache, glasses) [4].

JAFFE: The Japanese Female Facial Expression (JAFFE) database is developed in a laboratory-controlled environment. It is an image dataset that comprises of 213 image samples of posed expressions from 10 Japanese females. Each person has 3 to 4 images with each of six basic facial expressions and one image with a neutral expression [4].

There are five official folds in TFD; each fold consists of a training, validation, and test sets comprising of

TFD: The Toronto Face Database (TFD) is a combination of several facial expression datasets. TFD consists of 112,234 images, out of which 4,178 are annotated with one of the seven basic expression labels. There are five official folds in TFD; each fold consists of a training, validation, and test sets comprising of 70%, 10%, and 20% of the images, respectively [4].

DISFA: Denver Intensity of Spontaneous Facial Actions (DISFA) dataset comprises of video sequences belonging to 27 subjects. Each video sequence is recorded by two cameras while watching a four minutes video clip. This database is FACS coded with action unit (AU) intensity values [8].

FERA: It comprises of video recordings of 10 actors displaying a range of expressions. There are seven subjects in the training data, and six subjects in the test set. The training set constitutes 155 image sequences and the testing set constitutes 134 image sequences. The dataset has five emotion categories namely Anger, Relief, Happiness, Fear and Sadness [8].

SFEW: The Static Facial Expressions in the Wild (SFEW) database is created by selecting static frames from Acted Facial Expressions in the Wild (AFEW). It covers unconstrained facial expressions, age range, occlusions, various head poses, and close to real world illuminations. Overall, the dataset consists of 95 subjects. In total there are 663 labeled usable images [8].

FER2013: The database was created using the Google image search API and faces have been automatically registered. Faces are labeled as any of the six basic expressions as well as the neutral. The resulting database consists of 35,887 images most of them in wild settings [8].

2.2 Deep Facial Expression Recognition Systems

In this section, we briefly describe the state-of-the-art deep learning techniques for the facial expression recognition (FER) problem. Recently, deep learning techniques for FER problem has attracted a lot of research interest.

In order to improve the performance of neural network architectures, we mainly use two techniques, increasing the number of layers, increasing the number of neurons. This enables the network to learn more complex functions. The downside to increasing the complexity and depth of the networks is that it leads to number of issues like over-fitting of training data, and increased computational requirements [9]. One of the common solutions to the problem of dense networks is to create deep sparse networks. The authors in [10], provide a solution to these problems by providing an approximation to sparse networks. However, they retain the dense structure required for efficient computation.

In [11], the authors use a transfer learning approach for deep convolutional neural network (CNN) architectures. They initially use a network that has pre-trained on the ImageNet dataset, this is followed by fine tuning of the network in a two-step process, first on datasets relevant to facial expressions, followed by contest's dataset. The authors show that by cascading fine tuning approaches they are able to achieve better results. They achieve an overall accuracy of 48.5% in the validation set and 55.6% in the test set. The authors in [12] predict emotions in videos. They use a novel approach to extract temporal feature. The authors use recurrent neural network (RNN) to exploit temporal feature in videos along with spatial features extracted using convolution neural network (CNN). RNNs provide a framework to propagate information over a sequence using a continuous hidden layer representation. The authors show that their hybrid model of CNN-RNN based architecture outperforms CNN approach based on temporal averaging for aggregation.

The authors in [13], present a video-based emotion recognition based on a hybrid network. It combines recurrent neural network (RNN) and 3D convolutional networks (C3D) in a late-fusion fashion. Both these approaches encode appearance and motion information in different ways. RNNs take features extracted by convolutional neural network (CNN) over individual video frames as input and encodes motion later, whereas C3D model takes appearance and motion information of video simultaneously. Their framework achieved an accuracy of 59.02%.

The authors in [14] show that by replacing softmax layer with linear SVMs gives a significant performance improvement on FER2013 dataset. They achieve an accuracy of 71.2% on FER2013 dataset. In the work presented in [15], the authors present a bag of visual words model. It extracts dense SIFT descriptors from images. Then, it represents images as normalized presence vectors of visual words from a codebook obtained through clustering image descriptors and then local learning is used to predict class labels of test images. Their model achieved an accuracy of 67.49% on FER2013 dataset. The authors in [8] present a deep network that consists of two convolutional layers, each followed by max pooling and then four Inception layers. The network is a single component architecture that takes registered facial images as the input and classifies them into either the six basic expressions or the neutral expression. The authors approach reports an accuracy of 66.4% on FER dataset.

In the all these models, we need sufficient training examples belonging to each of the seven basic expressions for the network to perform well on the test dataset. Whereas in the current research work, we try to recollect new expressions for which there are few training examples. These training examples pertain to the previous seen instance.

2.3 Summary on One-shot learning approaches

In the recent times, there has been a significant increase in the use of One-shot learning for person re-identification problem. This is mainly owing to the fact that in real life scenario, it is very difficult to obtain labelled dataset for the person re-identification problem. In [16], input image pairs are partitioned into three overlapping horizontal parts respectively, and through a siamese CNN model to learn the similarity of them using cosine distance. The authors in [17] increase the depth of networks with using smaller convolution filters to obtain a robust feature.

In [18], the authors use few-shot learning for image recognition problem. They deal with a contrasting scenario where the number of categories is large and number of samples per each category is very limited in the dataset. Their framework has a novel approach where the model performs well for both large-scale and few-shot learning. Their idea is based on the observation that in the final classification layer of a pre-trained network, the activation vector and the parameter vector have highly similar structure in feature space. Their results demonstrate that their framework performance is comparable to the state-of-the-art classification accuracy on large scale ImageNet dataset and small miniImageNet dataset.

The authors in [19], present a Relation Network (RN) for few-shot learning. It is trained end-to-end from scratch. It learns an embedding and a deep non-linear distance metric for comparing query and sample items. The network is trained end-to-end based on episodic training that tunes the embedding and distance metric for effective few-shot learning. They show the effectiveness of the model on two datasets, Omniglot and miniImageNet. In [20], the authors use One-shot video object segmentation (OSVOS) for video object segmentation. Given the mask of the first frame, they separate an object from the background in a video in the successive frames. They perform experiments on two annotated video segmentation databases, and it shows that their framework is faster and their accuracy beats the state-of-the-art by a significant margin.

In the previous work, the concept of One-shot learning has been applied to person re-identification problem and to image recognition. In these two scenarios, there are many features available as we are dealing with entire image of person or the object. Whereas, in the current research work, the problem is more complex as we are looking at the subtle features in the face like widening and narrowing of eyes, widening of lips, changes in eyebrow shape, changes in forehead. Some of the expressions like surprised includes changes in more than one facial feature. But in real-life there are many subtle expressions that involves a subtle change in just one of the facial feature and this makes the problem even more challenging.

There has not been much work done towards One-shot learning for facial expression recognition (FER) problem. In the current research work, we initially train the neural network with FER2013 dataset and later use the features from the penultimate fully connected (FC) layer for One-shot learning. Using One-shot learning, we are able to recall unnamed new expressions with just one or very few training examples.

3. FACIAL EXPRESSION RECALLING PROBLEM

In this section, we explain in detail the facial expression recalling problem and motivation for choosing this research topic. We also give an overview of the techniques used to handle this problem.

Facial expression recalling problem has applications in various domains like education, business, and health-care. In education, for instance, if a teacher needs feedback from students whether they are understanding the material, it is possible to get their feedback without making them conscious of the examination. We can have few training samples or videos where the students find the material to be easily understanding and samples where they find it relatively hard to understand. We can extract the subtle facial expressions in both cases and use them as reference. Once we have the reference features, we can perform facial expression recalling in real-time to determine if the students have understood the material which is a very valuable asset to the academicians. They can adapt their teaching methods according to the students level of comprehending the material.

Facial expression recalling problem deals with recalling an expression that has been seen before. Suppose we see a facial expression and we have a history of previously seen facial expressions, we want to compare this expression with the previous ones to find out previous cases where same expression has appeared. The main challenge we face here is that there is no labelled data for the new expressions. The approach taken here is to use the existing neural network that is able to recognize the seven basic expressions and use the features extracted from the fully connected (FC) layer for One-shot learning.

First, we train the neural network to recognize named expressions. The neural network used in the framework is trained over FER2013 dataset and achieves an accuracy of 65%. In the current framework, the softmax loss function is used as the supervision signal to train the deep neural network (DNN). In order to enhance the discriminative power of the deeply learned features, the authors of [21] propose a new supervision signal, called center loss, for face recognition task. We use this technique in our framework for facial expression recognition problem. The center loss simultaneously learns a center for features of each class and penalizes the distances between their corresponding class centers and the features. We notice that the proposed center loss function is trainable and easy to optimize in the CNNs. With the joint supervision of softmax loss function along with center loss, we train the deep neural network (DNN) to obtain features with two key learning objectives, inter-class dispersion and intra-class compactness, that are of key importance to face expression recognition problem [21]. Performance of centerloss on the current framework has been summarized in section 4.

Our next step is to use the existing neural network to extract features for one-shot learning. One-shot learning is an object categorization problem. One-shot learning aims to learn information about object categories from one, or only a few training samples as against conventional machine learning based object categorization algorithms that require thousands of samples and large datasets. In facial expression recalling problem, since the number of training examples are very few, One-shot approach would be the best fit for the problem.

In the current framework, we perform 10-shot learning (since we use 10 images to learn information from the new expression set). In order to compare the features between the two images, we make use of two distance metrics Euclidean distance and Cosine similarity. They are the most common metrics used to compare the features. The authors in [22] introduce cosine similarity metric learning for face verification. We incorporate this approach in our framework to compare the features. The performance evaluation of the framework using each distance metric is summarize in section 4. We find empirically that cosine similarity greatly outperforms euclidean distance in the current framework.

Lastly, we extend the current research work to recognize expressions from 2D static images to 3D video clips. In 2D static images, we can only explore the spatial features using the neural network. But, in reality, each micro expression lasts for a period of 0.04s. So, along with spatial features, it would be beneficial if we could explore the temporal feature. Considering a 3D video clip instead of 2D static image will give us an added dimensionality in terms of temporal feature. This added dimension will help us to further improve the accuracy of One-shot learning in facial expression recalling problem.

Once the neural network has been tuned with the given dataset, we can then capitalize on the powerful discriminative features to generalize the predictive power of the network not just to new data, but to entirely new classed from unknown distributions [23]. In this research work, we use the features from the penultimate FC layer for the input image representation. These features have numerous advantages including, it gives a dimensionality reduction for the input images (48*48) to features representation of length (1024). Second, these features have a sparse representation in which most of the elements are equal to zero. Third, images that are similar are clustered together using this representation.

4. RESEARCH METHODOLOGY

In this section, we explain in detail the methodology used in the current framework.

4.1 Training Neural Network for Recognizing Named Expressions

We train a 11 layer deep convolution neural network over FER2013 dataset. The model is shown in 4.1. The input image is of size (48*48) which is fed as input to the neural network. The neural network architecture is defined as: CONV5-64, MAXPOOL5, CONV3-64, CONV3-64, AVGPOOL3, CONV3-128, CONV3-128, AVGPOOL3, FC1024, FC1024, SM7. The definition of each of the layer is as follows:

1. CONVw-n, 2D convolution layer with window size 'w' and 'n' number of filters.
2. MAXPOOLw, 2D max pool with window size 'w'
3. AVGPOOLw, 2D average pool with window size 'w'
4. FCn, fully connected layer of size 'n'
5. SMn, Softmax layer with 'n' outputs

The initial fully connected layers extract the low level features of the face, pooling helps to reduce the dimensionality. The later fully connected layers extract the high level features. Softmax layer helps to classify the input image into one of the seven different classes. The network is trained with FER2013 dataset.

4.1.1 The Center Loss

In order to have an effective loss function that improves the discriminative power of the deeply learned features, we use center loss function along with the existing softmax (or cross-entropy) loss functions as proposed in [24]. Using center loss helps to minimize the intra-class variations while keeping the features of different classes separable. The combined loss function (softmax

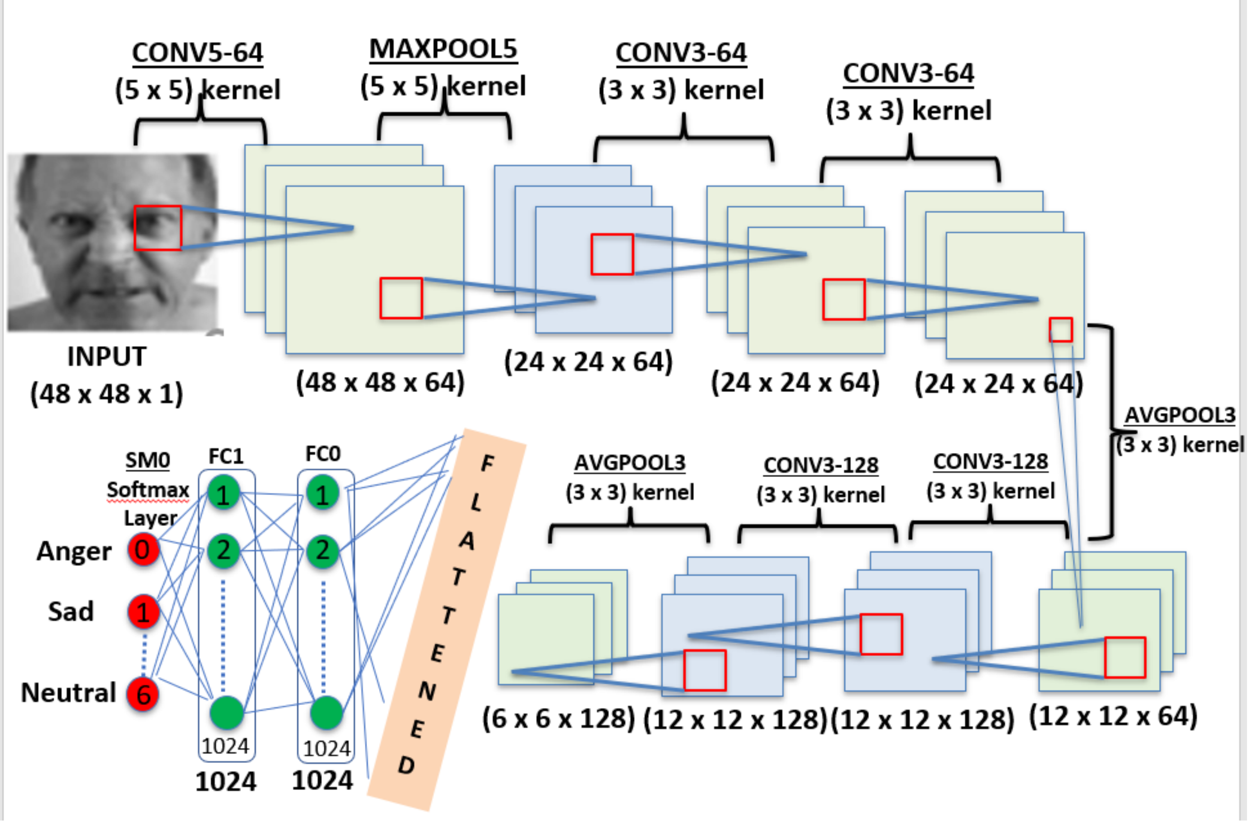


Figure 4.1: The architecture of CNN used for FER

loss and cross-entropy loss) is formulated as given in the equation below.

$$L_T = L_s + \lambda_c L_c$$

where L_T is the total loss that comprises of weighted sum of L_s, softmax loss (or cross-entropy loss) and L_c, center loss. λ_c is the weight corresponding to center loss. λ_c, is chosen as 0.008. This gives an optimal solution over other values of λ_c. Center loss, L_c is given by the expression below:

$$L_c = \frac{1}{2} \sum_{i=1}^{m} \|x_i - c_{y_i}\|_2^2$$

where x_i, is the input feature vector (which has a dimension of 1024). c_{y_i}, is the center of the y_i class of deep features.

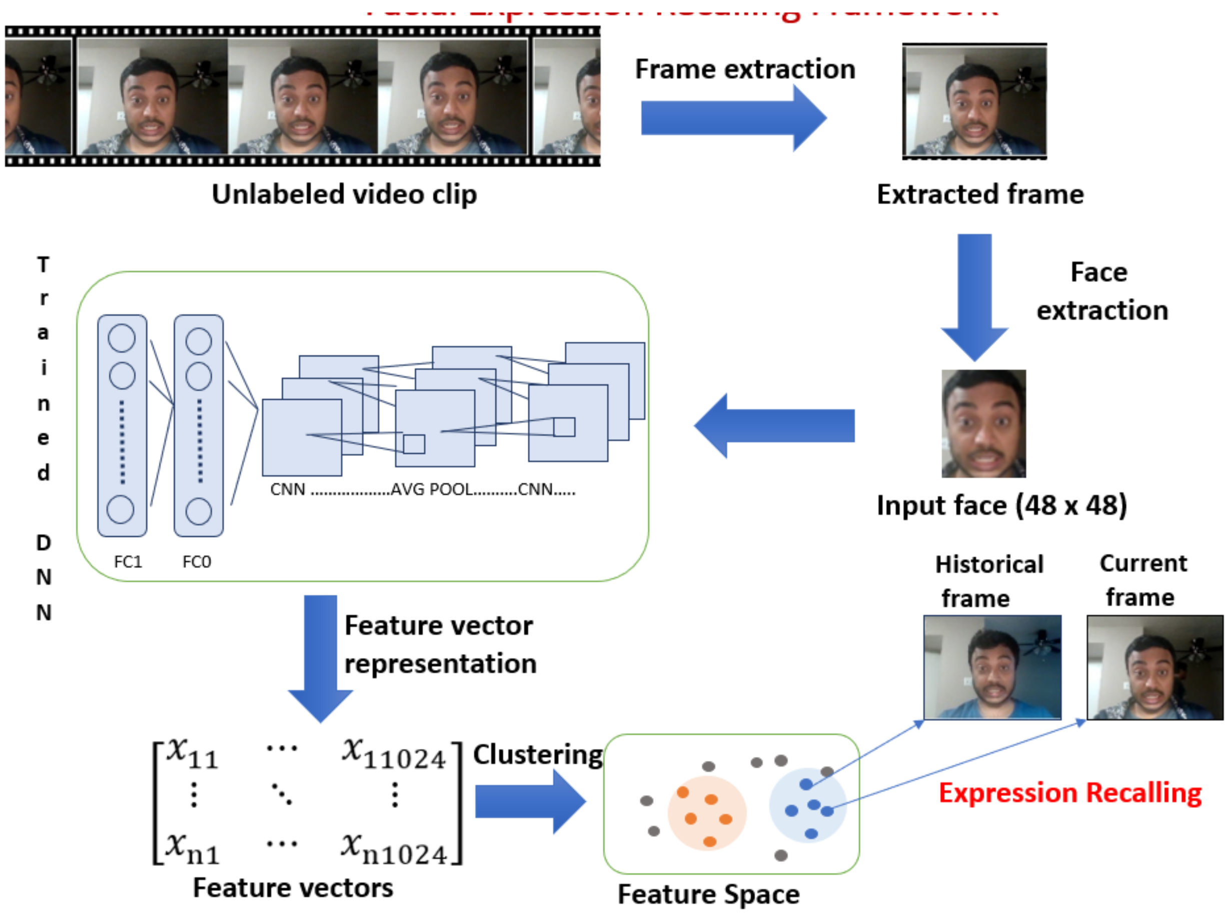

Facial Expression Recalling Framework
Frame extraction
Unlabeled video clip
Extracted frame
Face extraction
Trained DNN
CNN ....AVG POOL..........CNN.....
FC1
FC0
Input face (48 x 48)
Historical frame
Current frame
Feature vector representation
$\begin{bmatrix} x_{11} & \cdots & x_{11024} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{n1024} \end{bmatrix}$
Clustering
Expression Recalling
Feature vectors
Feature Space

Figure 4.2: Overview of Facial expression recalling problem

4.2 Overview of facial expression recalling framework

The flow chart for facial expression recalling problem is shown in 4.2. The input to the network is a 2D static image. We use Haar-cascade classifier to extract facial features from the image. The core basis for Haar classifier is the Haar-like features. These features use the change in contrast values between adjacent rectangular groups of pixels. The contrast variances between the pixel groups are used to determine relative light and dark areas. Two or three adjacent groups with a relative contrast variance form a Haar-like feature. Haar-like features are shown in 4.3.

Haar features can be scaled by increasing or decreasing the size of the pixel group being examined. This allows features to be used to detect objects of various sizes. But not all features are relevant. For instance, consider the face recognition problem shown in Fig. 4.4. The top row

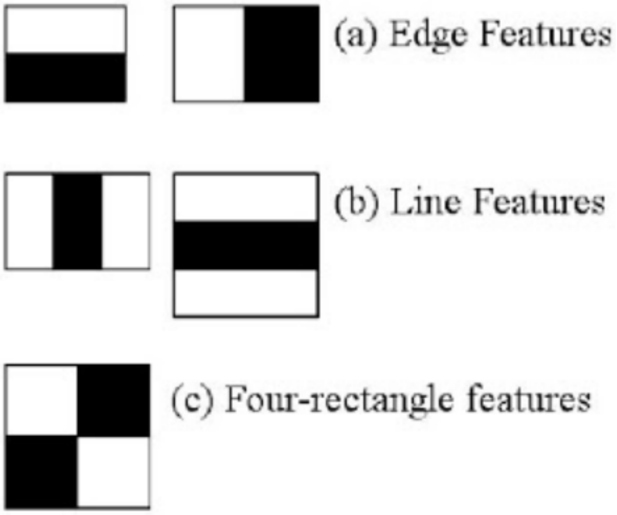


Figure 4.3: Haar-features [1]

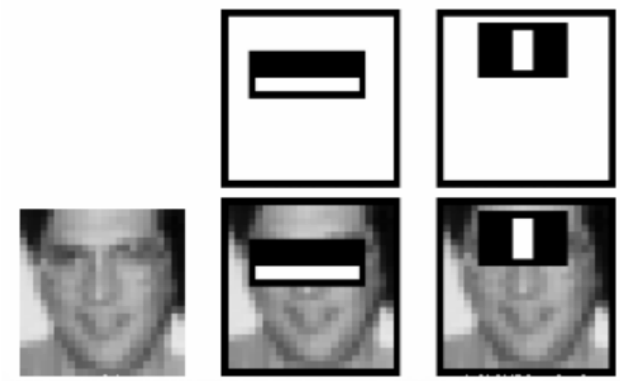


Figure 4.4: Haar-features for face recognition problem [1]

shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose [1].

Fig. 4.5 shows the flowchart of the face extraction framework using Haar-Cascade classifier. The input to the framework is the video frame that has a resolution of 1280x720 pixel at 15 fps. We use 'opencv2' library to process the video frame. As demonstrated in the flow chart we extract the individual frames from the video of dimension 1280x720 pixel. These frames are given as input to

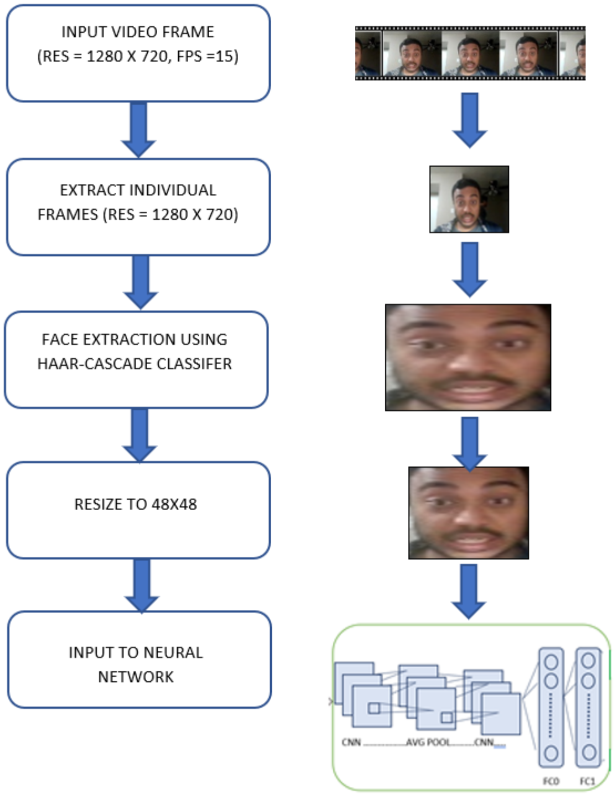


Figure 4.5: Flowchart of face extraction framework

the Haar-cascade classifier to extract the face from the image. But the extracted face is of a random dimension. In order to have a uniform dimension of the face across all the frames, we resize the image to 48x48 pixel dimension. This resized image is given as input to the neural network for training. Once the network has been trained with FER2013 dataset, we use the outermost 'dense1' layer of 1024 dimension as the feature vector for the input image representation. This feature vector is further used as input to One-shot learning in facial expression recalling problem.

4.3 Recalling Unnamed Expressions

Once the network has been trained, we use the features extracted from 'dense1' FC layer for One-shot learning. We initially extract the features from the reference video (or historical video) for each image frame and store the features. We later test our model with test video (or current video). Here again, we extract features of 1024 dimension for each image frame in test video and compare with previously stored features of the reference video. For comparing the two features, we use two distance metrics, Euclidean distance and Cosine similarity. These two metrics have been explained in detail below.

4.3.1 Euclidean distance and Cosine similarity

Let the two features be x and y. The Euclidean distance $E(x, y)$ between these two features is represented as follows.

$$E(x, y) = \|x - y\|_2$$

If the two features are similar then the Euclidean distance between the two points is close to zero otherwise it has a value far greater than zero.

On the other hand, Cosine similarity metric (CS) is defined as follows.

$$CS(x, y) = \frac{x^T y}{\|x\| \, \|y\|}$$

If the two vectors are close to each other, angle between the two vectors is close to zero. Thus Cosine similarity is close to one and the two images are similar. If the two vectors are far apart then Cosine similarity is close to zero and the two images considered are distinct.

4.3.2 Few-Shot learning using Leave One Subject Out

In this section, we introduce the concept of Few-Shot learning with Leave One Subject Out (LOSO). In Few-Shot learning, we make use of 10 train samples of the expression under test instead of all train images in the dataset. In Leave One Subject Out (LOSO), we train the expression with six out of seven expressions of FER2013 dataset and evaluate the performance of the left out seventh expression using Few-Shot learning. Once the network has been trained with six expressions, we extract the features of the penultimate fully connected layer for all the train samples belonging to the six expressions. For the left out seventh expression, we consider only 10 samples of this expression and the features for these samples are extracted.

Fig. 4.6 shows the overview of the Few-shot learning framework. Each cluster comprises

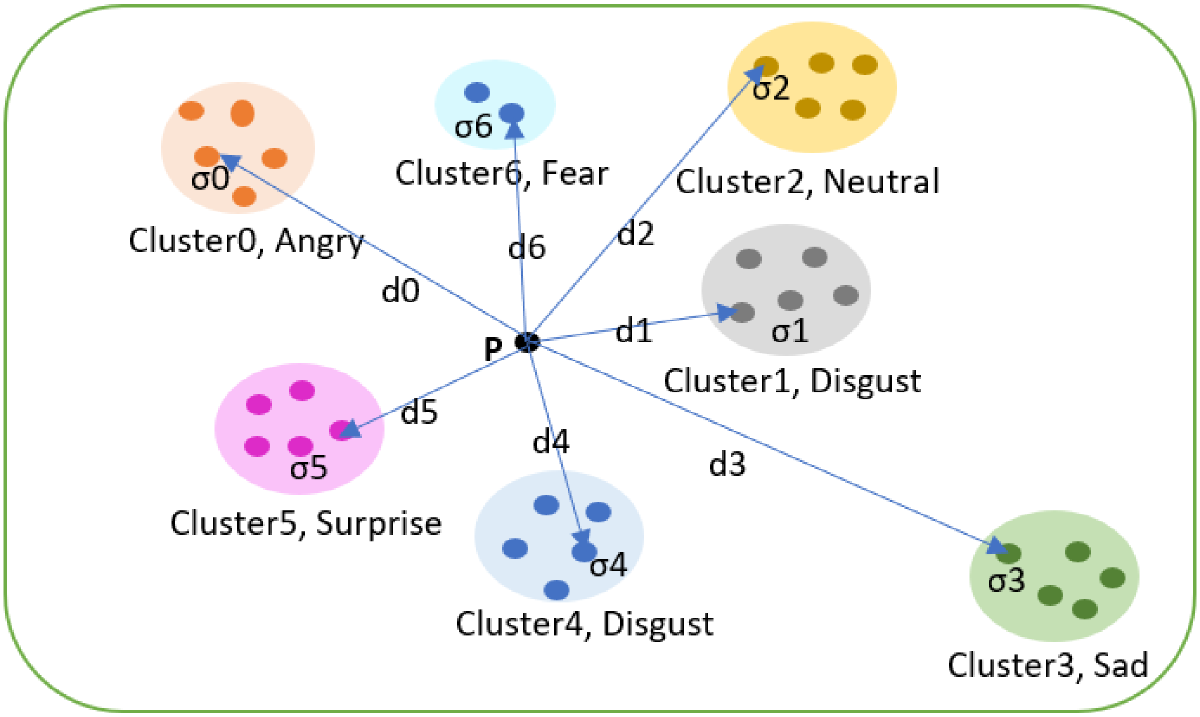


Figure 4.6: Overview of Few-Shot learning framework

of features extracted from the images. ‘Cluster0’, ‘Cluster1’, ‘Cluster2’, ‘Cluster3’, ‘Cluster4’, ‘Cluster5’ correspond to Angry, Disgust, Neutral, Sad, Disgust, and Surprise expressions. We train the neural network using these six expressions. For the left out expression (Fear) corresponding to ‘Cluster6’, we use only 10 samples of this expression. We then calculate the centroid for each cluster denoted by $\sigma 0$, $\sigma 1$, ... $\sigma 6$ corresponding to six different clusters. As shown in the figure, Let ‘P’ be feature vector of the test image. We then compute the distances ‘d0’, ‘d1’,...‘d6’ between the point of interest ‘P’ and the centroids of each cluster. We evaluate the performance using both the distance metrics, Euclidean distance and Cosine Similarity.

4.3.3 Distance band for recalling face expressions

In the section we introduce the concept of distance band that has been used in this framework. Fig. 4.7 shows the Euclidean and Cosine similarity distance band. Distance band is defined as the distance intervals for which the two frames (historical and current) are categorized

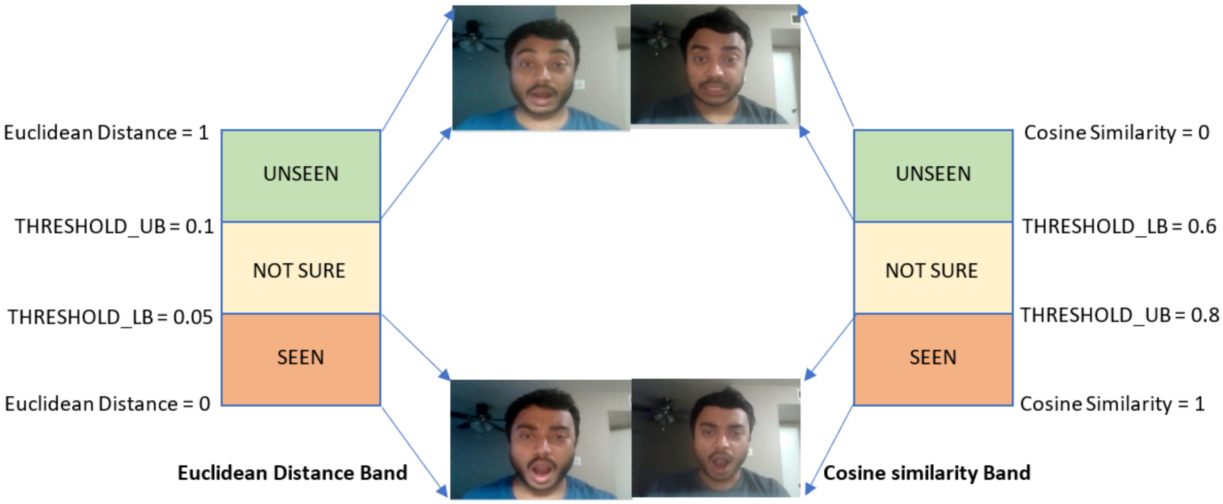


Figure 4.7: Euclidean and Cosine similarity distance band

into three different classes. The three classes are ‘SEEN’, ‘NOT SURE’, and ‘UNSEEN’. The boundaries of the intervals are defined by the threshold values. There are two types of thresholds, lower bound (‘THRESHOLD_LB’) and upper bound (‘THRESHOLD_UB’). In the context of Euclidean distance metric, let ED be the Euclidean distance between the two feature vectors, one feature vector corresponds to historical frame whereas the other feature vector corresponds to current frame. If $ED < THRESHOLD\_LB$, the two frames are categorized as ‘SEEN’, this indicates that the two expressions are similar. If $ED > THRESHOLD\_UB$, the two frames are categorized as ‘UNSEEN’, this indicates that the two expressions are distinct. If $THRESHOLD\_LB <= ED <= THRESHOLD\_UB$, the two frames are categorized as ‘NOT SURE’.

Similarly, in the context of Cosine Similarity, let CS be the cosine similarity between the two feature vectors. If $CS > THRESHOLD\_UB$, the two frames are categorized as ‘SEEN’. If $CS < THRESHOLD\_LB$, the two frames are categorized as ‘UNSEEN’. If $THRESHOLD\_LB <= CS <= THRESHOLD\_UB$, the two frames are categorized as ‘NOT SURE’. The values for the thresholds are tuned empirically based on the model performance.

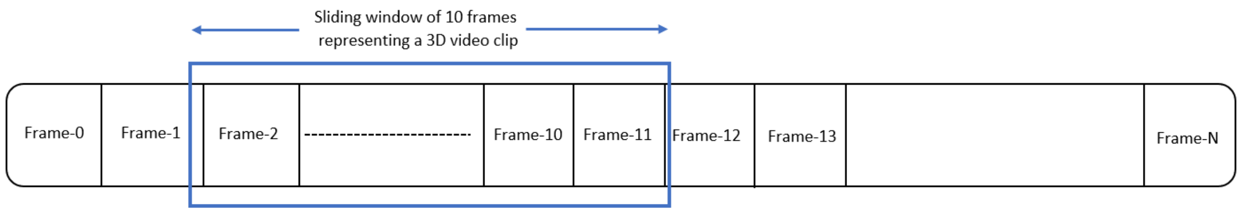


Figure 4.8: Sliding window approach for 3D video clip

4.3.4 k-means clustering of features

In order to evaluate the effectiveness of the features extracted by our current framework, we perform k-means clustering of features to analyze if all the features pertaining to a particular cluster correspond to similar expressions. k-means clustering aims to partition ‘n’ observations into ‘k’ clusters in which each observation belongs to the cluster with the nearest mean. Given a set of features $x1, x2, x3, ...xn$, where each observation is of 1024 dimensional real vector, k-means clustering aims to partition the ‘n’ observations into $k(<= n)$ sets $S = S1, S2, ...Sk$ so as to minimize the within-cluster sum of squares (WCSS).

4.3.5 Feature extraction in 3D video clip

In this section we explain the feature extraction for 3D short video clip. We use sliding window approach to extract the features for 3D video clip as show in Fig. 4.8. In this example we have considered 10 frames to represent each 3D video clip. However, we tune the number of frames for representation based on the model performance. In this approach, ‘frame0’ is represented as concatenated features of ‘frame0’, ‘frame1’,......‘frame9’. ‘frame1’ is represented as the concatenated features of ‘frame1’, ‘frame2’,....‘frame10’. However, by using this approach, we increase the dimensionality of frame representation from 1024 size to 10x1024. The added advantage is that we are now able to exploit temporal feature using this technique as against spatial features but the downside is the increase in the feature space dimension.

5. EXPERIMENTAL RESULTS

In this section, we summarize the results obtained in the current research work.

5.1 Training Neural Network

This section summarizes the training procedure and performance evaluation of the neural network. The neural network shown in Fig 4.1 is trained with FER2013 dataset. FER2013 dataset consists of 35,887 images that comprises of 28,709 train images, 3,589 validation images and 3,589 test images. It consists of images belonging to seven different expressions viz. angry, neutral, surprise, happy, fear, sad, and disgust as discussed in 2.1. The distribution of images across various expressions in the dataset is show in Table 5.1.

5.1.1 Pre-processing data and hyperparameter tuning

We use 'opencv2' library to convert color (RGB) input image to grayscale image. The input image is of size 48x48 = 2304 pixels. Prior to pre-processing data, we perform feature standardization of the input image data. By performing feature standardization, we achieve zero-mean and unit-variance of the input image data. Feature standardization is performed using the formula below.

$$X' = \frac{X - \mu}{\sigma}$$

Expression	Label	Total Images
Angry	0	4593
Disgust	1	547
Fear	2	5121
Happy	3	8989
Sad	4	6077
Surprise	5	4002
Neutral	6	6198

Table 5.1: FER2013 image distribution

Parameter	Value
Epochs	1000
Batch size	128
Optimizer	Adadelta
Learning Rate	0.1
rho	0.95
epsilon	1e-8
Loss function	cross-entropy loss along with centerloss

Table 5.2: Parameters chosen for training neural network

where μ is the mean of the features over the entire dataset and σ is the standard deviation. Once we have standardized features, we perform real-time data augmentation by applying random transformations including rotation, shifting images both vertically and horizontally, and randomly flipping images. This helps prevent overfitting and helps the model generalize better.

In order to select the best parameters for the model, we perform hyperparameter tuning through cross validation. The parameters selected for training the model are summarized in Table 5.2.

5.1.2 Train and test performance

The train and test accuracy with respect to epochs are shown in Fig 5.1. The training and test loss with respect to epochs is shown in Fig 5.2. The training accuracy at the end of 1000 epochs is 85.56% and test accuracy is 64.28%. The confusion matrix for test dataset is shown in the Table 5.3. The classification report is shown in Table 5.4. Since the number of images available for training across all the expressions is not uniform, the f1-score varies across different expressions. Table 5.1 shows the skewness in the training dataset of FER2013. The weighted average f1-score across all expressions is 0.64.

5.2 Evaluation of Few-Shot Learning for 2D static images

As explained in section 4.3.2, the Few-Shot learning for 2D static images is evaluated. The performance on the left out expression in Leave One Subject Out (LOSO) using the distance metrics (Euclidean Distance and Cosine Similarity) is summarized in Table 5.5. This table shows the confusion matrix for basic seven expressions. The comparison is made between baseline neural

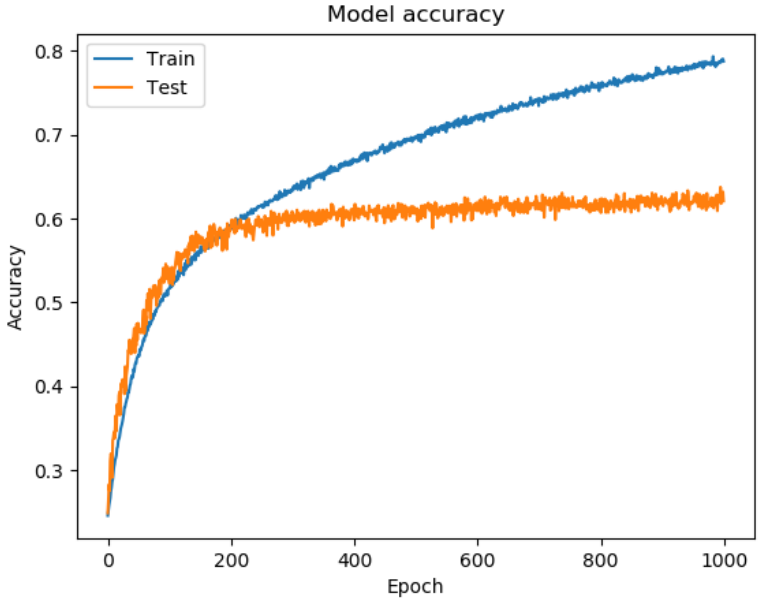


Figure 5.1: Model accuracy

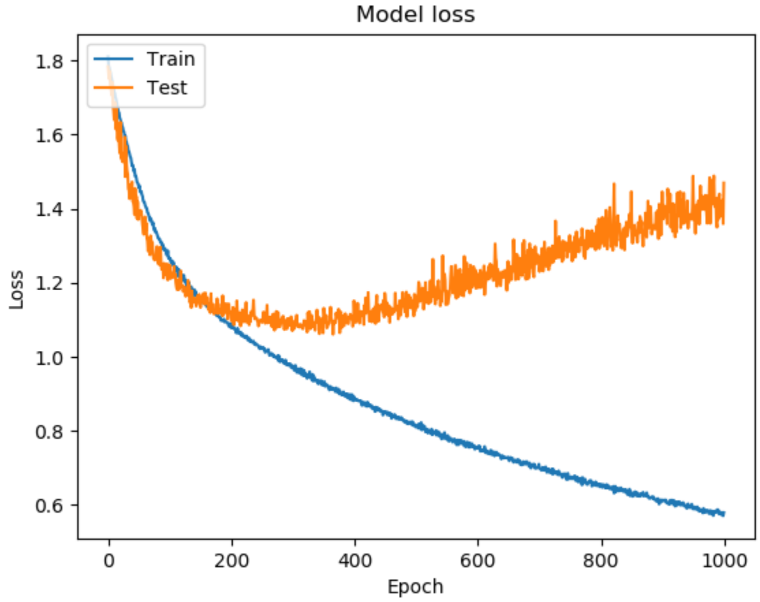


Figure 5.2: Model Loss

Expression	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	266	3	59	22	76	8	56
Disgust	14	30	3	3	4	0	1
Fear	78	2	241	10	99	41	57
Happy	38	0	20	740	24	10	47
Sad	62	3	66	26	302	7	128
Surprise	20	0	42	22	6	309	16
Neutral	28	1	33	32	107	9	416

Table 5.3: Confusion matrix for all seven expressions on test dataset

Expression	Precision	Recall	f1-score	Samples
Angry	0.53	0.54	0.53	490
Disgust	0.77	0.55	0.64	55
Fear	0.52	0.46	0.49	528
Happy	0.87	0.84	0.85	879
Sad	0.49	0.51	0.50	594
Surprise	0.80	0.74	0.77	415
Neutral	0.58	0.66	0.62	626
Weighted Avg.	0.65	0.64	0.64	3587

Table 5.4: Classification report

network model that has been trained with all seven expressions (highlighted in ‘cyan’). This represented as the baseline in the table. For every expression, second row represents the result using Leave One Subject Out (LOSO) for Euclidean Distance (ED), the third row represents the result for Cosine Similarity (CS).

For instance, consider the first expression, ‘Angry’. In Leave One Subject Out, the network is trained with the trained dataset of the remaining six expressions excluding ‘Angry’ expression. Then as explained in section 4.3.2, the features are extracted for all the train images of the remaining six expressions. Whereas, for ‘Angry’ expression features corresponding to only ten images are extracted. Here the features are obtained for the penultimate fully connected (FC) layer of 1024 dimension. Then the centroids corresponding to each of the seven clusters are computed. In order to classify the image under test, we extract the features and compare the performance using two distance metrics ‘Euclidean Distance’ and ‘Cosine Similarity’.

5.3 Effectiveness of 3D video clip over 2D static image

In this section we evaluate the performance of feature extraction of 3D video clip over 2D static images. We perform evaluation on the video consisting of subtle and compound expressions. Some of the expressions include ‘movement of eyeball’, ‘sadly surprised’, ‘understanding’, ‘not-understanding’, ‘angrily surprised’, ‘awestruck’, ‘happily surprised’, ‘sadly fearful’. These expressions are very different from the seven basic expressions. We perform the experiment in two phases. First, we extract the features for 2D static images and then perform k-means clustering as explained in section 4.3.4.

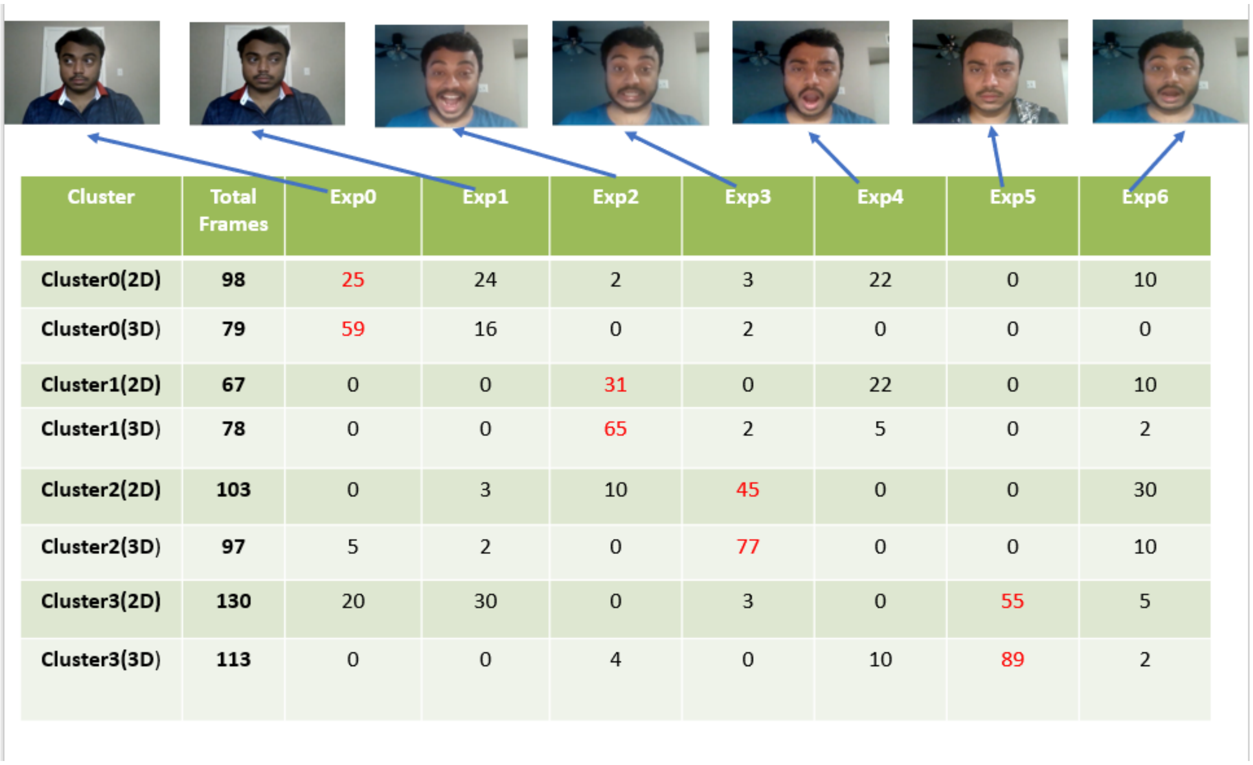
Second, we extract the features for 3D short video clip and perform k-means clustering. As explained in section 4.3.5, we extract features from contiguous frames instead of a single frame through sliding window approach. The results are shown in Fig. 5.3. We notice that 3D short video clip performs better than 2D static images. This is owing to the fact that, 3D short video clip exploits temporal feature. This is in addition to the spatial features extracted through deep neural network (DNN). Whereas, with 2D static images only spatial features are extracted and temporal feature is not considered. So, considering a single 2D static image for expression recalling would not give us very accurate results.

As we notice from the previous results, when we use 2D static images, some of the new expressions are mis-classified. This is owing to the fact that a human micro expressions lasts for a duration of 0.04s. So, considering a single 2D static image for expression recalling would not give us very accurate results. We further extend this method to 3D video clips. As explained in section 4.3.5, we extract features from contiguous frames instead of a single frame through sliding window approach. We evaluate the performance of the two distance metrics ‘Euclidean Distance’ and ‘Cosine Similarity’ for face expression recalling of 3D short video clips. The results are tabulated in Fig. 5.4.

As explained in section 4.3.3, we empirically find the threshold values for both the distance metrics. The ‘THRESHOLD_LB’ for ‘Euclidean Distance’ is 0.07 and ‘THRESHOLD_UB’ is

Expression	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry (baseline)	233	4	35	32	104	9	50
Angry (10-Shot, ED)	93	71	86	80	57	33	47
Angry (10-Shot, CS)	69	82	96	90	60	31	39
Disgust (baseline)	21	14	6	3	9	0	3
Disgust (10-Shot, ED)	17	17	4	7	8	0	0
Disgust (10-Shot, CS)	15	23	3	5	6	4	0
Fear (baseline)	70	1	148	37	146	39	55
Fear (10-Shot, ED)	61	138	81	110	24	51	31
Fear (10-Shot, CS)	73	105	147	54	46	62	9
Happy (baseline)	37	0	25	693	66	28	46
Happy (10-Shot, ED)	82	63	109	323	261	29	28
Happy (10-Shot, CS)	102	103	135	264	232	46	13
Sad (baseline)	86	2	60	55	353	8	89
Sad (10-Shot, ED)	74	84	174	58	169	21	73
Sad (10-Shot, CS)	92	124	145	65	189	31	7
Surprise(baseline)	14	0	43	22	22	301	13
Surprise (10-Shot, ED)	24	85	40	43	19	194	10
Surprise (10-Shot, CS)	45	138	26	49	45	111	1
Neutral (baseline)	51	0	45	56	134	11	310
Neutral (10-Shot, ED)	86	92	122	64	14	24	205
Neutral (10-Shot, CS)	72	5	122	67	33	1	227

Table 5.5: Confusion matrix - comparison of baseline NN with 10-shot learning



Cluster	Total Frames	Exp0	Exp1	Exp2	Exp3	Exp4	Exp5	Exp6
Cluster0(2D)	**98**	25	24	2	3	22	0	10
Cluster0(3D)	**79**	59	16	0	2	0	0	0
Cluster1(2D)	**67**	0	0	31	0	22	0	10
Cluster1(3D)	**78**	0	0	65	2	5	0	2
Cluster2(2D)	**103**	0	3	10	45	0	0	30
Cluster2(3D)	**97**	5	2	0	77	0	0	10
Cluster3(2D)	**130**	20	30	0	3	0	55	5
Cluster3(3D)	**113**	0	0	4	0	10	89	2

Figure 5.3: Effectiveness of 3D short video clip over 2D static images

0.10. Similarly, for ‘Cosine Similarity’, ‘THRESHOLD_LB’ is 0.6 and ‘THRESHOLD_LB’ is 0.8. As shown in Fig. 5.4, using Cosine Similarity, we can detect even subtle expressions like movement in eyeball and it also performs better with other expressions compared to Euclidean Distance.

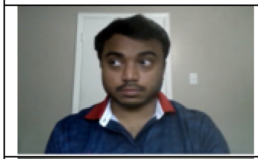
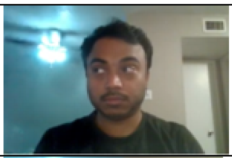
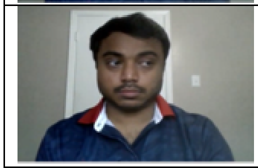
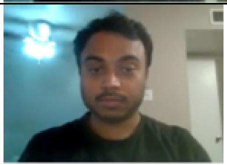
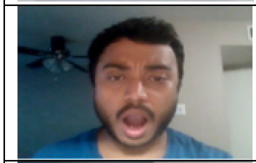
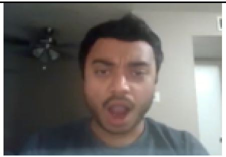
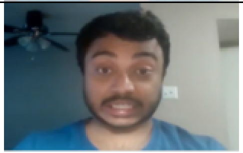
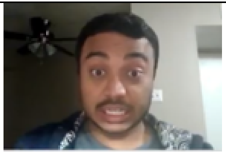
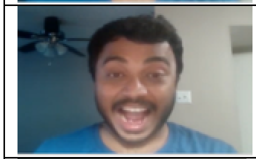
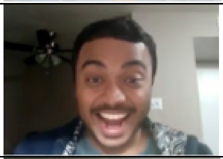
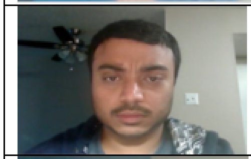
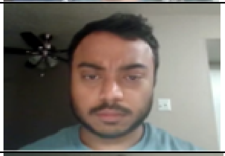
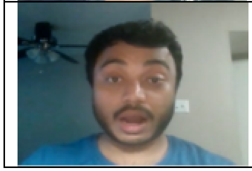
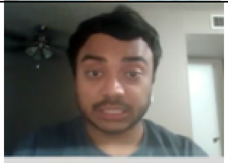
Current video	Historical video	Euclidean distance	Expression recalling with Euclidean distance (THRESHOLD_LB = 0.07, THRESHOUL_UB = 0.10)	Cosine similarity	Expression recalling with Cosine similarity (THRESHOLD_LB = 0.6, THRESHOUL_UB = 0.8)
		0.13	UNSEEN	0.84	SEEN
		0.15	UNSEEN	0.83	SEEN
		0.06	SEEN	0.9	SEEN
		0.07	NOT SURE	0.93	SEEN
		0.06	SEEN	0.91	SEEN
		0.12	UNSEEN	0.87	SEEN
		0.17	UNSEEN	0.81	SEEN

Figure 5.4: Evaluation of face expression recalling using 3D video clips

6. CONCLUSION AND FUTURE WORK

From the results obtained in the previous section, we conclude that we are able to recall new unnamed expressions using the features obtained from the neural network trained over named expressions. The performance improves significantly when we consider short 3D video clips instead of 2D static images. This is mainly because 3D video clip captures temporal information in addition to the spatial information captured by the CNN. This additional dimension in the input features helps to improve the accuracy of recalling new unnamed expressions.

In the future, we would like to further extend our work to 3D CNN implementation for video clips. Second, we would like to explore the hybrid network implementation involving CNN and LSTM networks for facial expression recalling. With this hybrid network, we can explore temporal feature through LSTM and spatial feature through CNN. Third, we would like to reduce the feature space dimension for representing 3D video clip. This would reduce the memory requirement for representing 3D short video clip. Lastly, we would like to embed the audio signals with Natural Language Processing (NLP) with current facial expression recalling framework. This would help us to better understand the context.

REFERENCES

[1] "Opencv documentation on haar-cascade classifiers." Web.

[2] C. C. Chibelushi and F. Bourel, "Facial expression recognition: A brief tutorial overview," *On-Line Compendium of Computer Vision*, 2003.

[3] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion," *Journal of personality and social psychology*, vol. 17, pp. 124–129, 1971.

[4] S. Li and W. Deng, "Deep facial expression recognition: A survey," *CoRR*, vol. abs/1804.08348, 2018.

[5] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, and D. Thaler, "Challenges in representation learning: A report on three machine learning contests," *International Conference on Neural Information Processing*, 2013.

[6] C. Pramerdorfer and M. Kampel, "Facial expression recognition using convolutional neural networks: State of the art," *CoRR*, vol. abs/1612.02903, 2016.

[7] "Which face is 'happily disgusted'? scientists discover that humans have 21 different facial expressions." Web.

[8] A. Mollahosseini, D. Chan, and M. Mahor, "Going deeper in facial expression recognition using deep neural networks," *IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE*, 2016.

[9] A. Mollahosseini *et al.*, "Going deeper in facial expression recognition using deep neural networks," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[10] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *arXiv*, 2014.

[11] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, "Deep learning for emotion recognition on small datasets using transfer learning," in *Proceedings of the 2015 ACM on Interna-*

tional Conference on Multimodal Interaction, ICMI '15, (New York, NY, USA), pp. 443–449, ACM, 2015.

[12] S. Ebrahimi Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal, "Recurrent neural networks for emotion recognition in video," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, (New York, NY, USA), pp. 467–474, ACM, 2015.

[13] Y. Fan, X. Lu, D. Li, and Y. Liu, "Video-based emotion recognition using cnn-rnn and c3d hybrid networks," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ICMI '16, (New York, NY, USA), pp. 445–450, ACM, 2016.

[14] Y. Tang, "Deep learning using support vector machines," *CoRR*, vol. abs/1306.0239, 2013.

[15] R. Ionescu, M. Popescu, and C. Grozea, "Local learning to improve bag of visual words model for facial expression recognition," *Workshop on Challenges in Representation Learning, ICML*, 2013.

[16] D. Yi, Z. Lei, S. Liao, S. Z. Li, *et al.*, "Deep metric learning for person re-identification," *ICPR*, 2014.

[17] L. Wu, C. Shen, and A. van den Hengel, "Personnet: Person re-identification with deep convolutional neural networks," *CoRR*, vol. abs/1601.07255, 2016.

[18] S. Qiao, C. Liu, W. Shen, and A. L. Yuille, "Few-shot image recognition by predicting parameters from activations," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[19] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[20] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[21] Y. Wen, K. Zhand, and Y. Qiao, “A discriminative feature learning approach for deep face recognition,” *ECCV*, 2016.

[22] H. Nguyen and L. Bai, “Cosine similarity metric learning for face verification,” *ACCV*, 2010.

[23] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese neural networks for one-shot image recognition,” *ICML Deep Learning workshop*, 2015.

[24] Y. Wen, K. Zhang, and Y. Qiao, “A discriminative feature learning approach for deep face recognition,” *ECCV*, 2016.