An Investigation into Modern Facial Expressions Recognition by a Computer

by

Sachin Chhabra

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2019 by the
Graduate Supervisory Committee:

Baoxin Li, Chair
Hemanth Venkateswara
Siddharth Srivastava

ARIZONA STATE UNIVERSITY

May 2019

ABSTRACT

Facial Expressions Recognition using the Convolution Neural Network has been actively researched upon in the last decade due to its high number of applications in the human-computer interaction domain. As Convolution Neural Networks have the exceptional ability to learn, they outperform the methods using handcrafted features. Though the state-of-the-art models achieve high accuracy on the lab-controlled images, they still struggle for the wild expressions. Wild expressions are captured in a real-world setting and have natural expressions. Wild databases have many challenges such as occlusion, variations in lighting conditions and head poses. In this work, I address these challenges and propose a new model containing a Hybrid Convolutional Neural Network with a Fusion Layer. The Fusion Layer utilizes a combination of the knowledge obtained from two different domains for enhanced feature extraction from the in-the-wild images. I tested my network on two publicly available in-the-wild datasets namely RAF-DB and AffectNet. Next, I tested my trained model on CK+ dataset for the cross-database evaluation study. I prove that my model achieves comparable results with state-of-the-art methods. I argue that it can perform well on such datasets because it learns the features from two different domains rather than a single domain. Last, I present a real-time facial expression recognition system as a part of this work where the images are captured in real-time using laptop camera and passed to the model for obtaining a facial expression label for it. It indicates that the proposed model has low processing time and can produce output almost instantly.

ACKNOWLEDGMENTS

I would like to express my gratitude to Dr. Baoxin Li for his guidance and support throughout the project. I thank him for patiently encouraging me especially in times when I was stuck with my experiments. Furthermore, I would like to thank him for preparing me to gain expertise in the areas of Computer Vision and Deep Learning.

I would also like to thank my committee members Dr. Hemanth Venkateshwara and Dr. Siddharth Srivastava for their valuable time for being a part of my defense committee.

In addition, I would like to thank my lab mates and senior Ph.D. students, Mr. Kevin Ding and Miss. Yuzhen Ding for helping me with my baseline experiments. Also, the deep learning tips and tricks taught by them proved to be quite useful in my experiments.

Lastly, I would like to thank all my lab-mates for maintaining a work-friendly environment in the lab and providing help whenever needed.

TABLE OF CONTENTS

Page

LIST OF TABLES vi

LIST OF FIGURES vii

CHAPTER

1 INTRODUCTION 1

1.1 Motivation 1

1.2 Problem Statement 3

1.3 Related Work 4

1.3.1 Handcrafted Features for Recognition 4

1.3.2 Deep Learning Based Models 5

1.4 Contributions 6

1.4.1 Fusion Layer 6

1.4.2 Cross-database Study 6

1.4.3 Real-time Facial Expression Recognition 7

1.4.4 Specific Contributions 7

2 DATASETS AND BASELINE MODELS 9

2.1 Data-sets for Facial Expression Recognition 9

2.1.1 Raf-db Dataset 10

2.1.2 Affectnet Dataset 11

2.1.3 CK+ Dataset 13

2.2 Problems with In-the-wild Datasets 14

2.2.1 Occlusion 15

CHAPTER Page

2.2.2 Head pose……..…15

2.2.3 Illumination……..……16

2.2.4 Class Imbalance....……16

2.3 Baseline Models 17

2.3.1 Imagenet……..17

2.3.2 Facenet……18

2.3.3 Vgg-16 Network ……....…18

2.4 Metric Used for Comparison.....19

3 PROPOSED METHOD-FUSION LAYER 21

3.1 Hybrid Neural Network with Fusion Layer.....21

3.2 Model Architecture.....22

3.3 Experiments and Results 25

3.3.1 Experiments on Raf-db Dataset…….……..26

3.3.2 Experiments on Affectnet Dataset ….……28

3.3.3 Experiments on Alpha…….……30

3.3.4 Experiments on Ck+ Dataset ……...……..30

3.4 Analysis of the Results 32

4 IMPLEMENTING A REAL-TIME DEMO SYSTEM.. 34

4.1 Practical Considerations in Building a Real-time System…….……34

4.1.1. Real-time Video Streaming…….…….……34

4.1.2. Face Detection…….…….……35

4.1.3. Face Alignment…….……38

CHAPTER Page

4.1.4. Resizing ……..……40

4.1.5. Handling Varying Illumination Conditions…….……41

4.1.6 Predictions.…….……..…….……42

4.2. Sample Demo Implementations 42

5 CONCLUSION 49

REFERENCES..... 52

LIST OF TABLES

Table Page

1. Confusion Matrix for Raf-db Dataset 27
2. Comparison of Different Models on Raf-db Dataset 28
3. Confusion Table for Affectnet Data 29
4. Comparison of Domain Fusion to Individual Domains on Affectnet Dataset 30
5. Confusion Table for Ck+ Dataset Using Model Trained on Raf-db Dataset. 31
6. Confusion Table for Ck+ Dataset Using Model Trained on Affectnet Dataset. . 32

LIST OF FIGURES

Figure Page

1. Facial Expression Recognition Framework 1
2. Sample Image from Each Class of Raf-db Dataset..... 10
3. Sample Image from Each Class of Affectnet Dataset..... 12
4. Sample Image from Each Class of Ck+ Dataset.....14
5. Problems Encountered While Using In-the-wild Dataset ……..15
6. Architecture of Vgg-16 Network ……..19
7. Proposed Model Architecture ……22
8. Channel Mapping ……...……....23
9. Fusion Layer ……..…….24
10. Capturing the Frame from Real-time Streamed Video ……....35
11. Detecting the Face from the Captured Frames …….……...36
12. Comparison of Viola-jones and Dlib Algorithm …….……....37
13. 68 Facial Landmarks given by the Dlib Algorithm ..…...…..……...38
14. Facial Landmarks to Perform Face Alignment Step .…….……..40
15. Resized and Cropped Image…….....…….……..…….....40
16. Image after Histogram Equalization ….…….…….…….42
17. Experimental setup of real-time facial expression recognition ..…….43
18. Example of Facial Expression Real-time Demonstration...……44
19. Output by the Model for Anger Expression ....…….……....45

Figure Page

20. Output by the Model for Fear Expression ……..……...45

21. Output by the Model for Disgust Expression ……...….……...46

22. Output by the Model for Happy Expression …….…….……..46

23. Output by the Model for Neutral Expression …….…….…..…47

24. Output by the Model for Sad Expression …….…….……..….47

25. Output by the Model for Surprise Expression ….……...……..…..48

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

Facial expressions are the non-verbal communication to understand the mental state of a human being. Even though nothing much is said, there is a lot to comprehend the messages. People often communicate just through expressions. Recognizing these expressions is an easy job for a human but it is a big challenge for a computer. Understanding the expressions can have applications in various domains like Human Computer Interaction, video games, call centres, etc. Expressions can be classified into six basic categories - anger, disgust, fear, happiness, sadness, surprise or seven including the neutral expression [5].

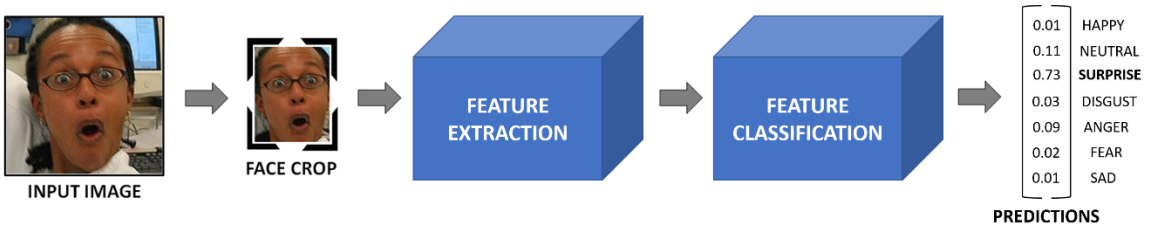

Figure 1: Facial expression recognition framework. The first step is extracting the face from the original image and eliminate the background. These face crops are fed to the Feature extractor followed by Feature classification. At the output, the class label according to the predicted probabilities for the image is obtained.

Facial expression recognition can be divided into three steps - localizing face, followed by feature extraction and expression classification [7]. The framework is displayed in

Figure 1. The first step includes identifying and extracting the face from the images to get rid of the background. For feature extraction, the earlier approaches used to handcraft the features based on appearance or geometry like HOG filters [1], LBP [21], GABOR filters [13], etc. After the success of AlexNet [10] at ImageNet Large scale visual Recognition Challenge, the recent approaches include the use of a convolutional neural network to learn the features, and the last step is the classification of features using a classifier like support vector machine (SVM), cross entropy, K-Nearest Neighbor (KNN), etc.

CNNs use the softmax probabilities at the output layer which combines the feature extraction and classification step into one model. They are now the go-to approach to solve the computer vision problems replacing the handcrafted features, but they suffer the problem of over-fitting in case of facial expressions due to lack of enough training data. This demands more data-augmentation/transfer learning techniques [27]. Most of the facial expressions' datasets like CK+ [14], JAFFE [15], etc. are small, captured in lab settings and lack the natural human responses, natural settings, etc. There are some in-the-wild datasets [32] which are getting popular these days. They collect images from the internet or movies and are relatively larger in size. Recognizing expression here gets more challenging as these images suffer from different lighting conditions, occlusions, head-pose, blur and demands preprocessing of the data. Additionally, they have high intra-class variations due to wide-ranging personal features like age, gender, ethnicity, etc. Also, the classes of the wild datasets are highly imbalanced, as it is easier to get pictures of some classes than others. For example, it is quite easy to obtain a picture of happy expression than the fear expression. As a result of the class imbalance, the network creates a bias on the class having a high number of classes during prediction. Due to these issues, classifying

facial expressions for in-the-wild images is a bigger challenging task than on the lab-controlled ones.

1.2 PROBLEM STATEMENT

To address the above issues, we propose a hybrid neural network that uses transfer learning from two neural networks trained on two different datasets to extract the features. A lot of recent work for facial expression recognition use transfer learning [33], [34], [35] and source domain plays a huge role in the target performance. A general approach is to use a network trained on ImageNet [2] dataset as the network is a good general feature extractor. Though it performs well on ImageNet dataset, it's performance decreases on facial expressions' dataset as it cannot extract the features of images containing faces properly.

Recent works are using networks trained on the face recognition datasets for face recognition task [19]. A network trained on FaceNet dataset extracts the excellent features as the source domain is closer to the target domain. The network performs well on the lab-controlled datasets but fails on in-the-wild datasets. This is because the in-the-wild images have issues mentioned in the previous section.

Therefore, we propose a hybrid neural network where two VGG-16 [22] networks trained on ImageNet and FaceNet datasets are merged to extract the deep features together. The network trained on FaceNet will extract the features of the faces while the network trained on ImageNet extract the general features. A combination is necessary as the in-the-wild images do not contain just frontal upright faces as in lab-controlled databases. We use RAF-DB [12] and AffectNet [16] datasets to showcase the performance of my approach.

Next, to study the transferability of the features, I test the trained models on CK+ dataset without any fine-tuning. This is to showcase that the network trained on in-the-wild dataset works well on lab-controlled dataset while the vice-versa is not true. This demonstrates the generalization ability of the model.

To further show that my model can run fast and give accurate results in real-time, I demonstrate the real-time facial expression recognition using a web camera of the laptop. The face is detected from the captured frames. To handle problems like non-uniform lighting and varying head poses [37], various algorithms are applied so that the image is ready to be passed as input to the proposed model. The model will predict facial expression and display it on the screen.

1.3 RELATED WORK

1.3.1 Handcrafted features for Facial Recognition:

Recognizing facial expression has been an active study for decades. Earlier approaches were based on using hand-crafted filters to extract features from the images. Handcrafted filters are generally of two types - geometry based and appearance based. Geometry based features extract the angle, distance and shape features between the facial points like lips, eyes, nose, etc. The appearance-based features apply a filter on the whole/sub-regions of the image to extract texture information. [17] used geometry-based features to classify expressions. They used gradients of radial symmetry for points at a fixed distance in the direction of the gradient, orientation projection and a magnitude projection of the image to locate eyes, nose, and mouth in the face. The distance of eyebrows, the distance between the right eyebrow and nose tip, the distance between the left eyebrow and nose tip, mouth

width and mouth height for the face were used as the features for classification of the expressions.

1.3.2 Facial Recognition using Deep Learning Models:

After the success of AlexNet in the ImageNet Large Scale Visual Recognition Challenge, deep learning has been widely used for computer vision tasks. [25] addressed one of the earliest approaches to solve the facial expression recognition problem using a convolutional neural network. They showcased the use of a Support Vector Machine (SVM) for deep learning. Their model used a simple convolutional network and a linear One vs All SVM to train the model instead of cross-entropy loss and reported an increase in the accuracy. [28] used a pre-trained CNN model on FER 2013 Challenge data-set and fine-tuned it on SFEW dataset. They trained multiple CNN models in the same way. Instead of averaging the output, they were arranged in the ensemble way and the weights of the different models were learned using ensemble log-likelihood loss. [18] shows the use of transfer learning between datasets. They used a pre-trained network on ImageNet and fine-tuned it on FER2013 [24] facial expression dataset prior to training on the target SFEW [3] dataset. The two-step fine-tuning outperformed direct single training.

[12] introduced a new convolutional neural network architecture for facial expression recognition and locality preserving loss that pulls the k neighbors of the same classes together. It increases the inter-class distance and reduces the intra-class variations when trained along with cross-entropy loss function. [4] used a network trained for facial recognition task to extract the deep facial features to classify expressions. [11] claimed a small convolutional neural network model is enough for expression recognition. Their

model had very few parameters compared to traditional CNNs and was able to classify facial expressions correctly. [8] used VGG-face extracted deep features along with handcrafted features computed by the bag-of-visual-words (BOVW) model. [6] proposed a network that utilizes a pre-trained VGG network on ImageNet to extract features of the faces from different sub-regions, combined with features of the global face in a weighted manner to do the classification.

1.4 CONTRIBUTIONS

1.4.1 Fusion Layer:

In this work, I propose a hybrid network with a layer called Fusion Layer. This layer utilizes a combination of knowledge from two different domains. As it uses a combination, the feature extracted for the in-the-wild datasets is enhanced. The proposed network is tested on RAF-DB and AffectNet datasets. The state-of-the-art methods for facial expression recognition work well on lab-controlled datasets. In order to achieve such performance on in-the-wild datasets, this fusion layer approach is proposed. It gives comparable results on the mentioned datasets.

1.4.2 Cross-Database Study:

Apart from in-the-wild datasets, the proposed model performs exceptionally well on the lab-controlled database like CK+. When the model trained on in-the-wild images is tested on CK+ directly, it achieves high accuracy. This suggests that the approach works well for both the lab-controlled as well as in-the-wild datasets, unlike the state-of-the-art methods for lab-controlled datasets, which fail to perform well on in-the-wild datasets.

1.4.3 Real-time Facial Expression Recognition:

To show that my proposed model works on images taken in real-time, a real-time demo for the same is presented. The prediction made by the model on the captured frame is almost instant which suggests that my proposed model can work on a real-time system with a high speed as well. This real-time demo can handle cases when there is no face for certain consecutive frames.

1.4.4 Specific Contributions:

Below are my contributions through this work:

1. A novel method to combine knowledge from two different domains is proposed. It proves to be very useful while performing facial expression classification task on in-the-wild datasets. No such prior work is done to best of my knowledge.
2. It is shown that the proposed model performs extremely well on in-the-wild datasets and gives comparable results with the state-of-the-art methods.
3. Apart from in-the-wild datasets, the hybrid network performs well on the lab-controlled dataset also. This cross-database study indicates that the network trained on in-the-wild datasets performs well on the lab-controlled datasets.
4. A real-time expression recognition is demonstrated to show the speed and accuracy of the proposed model on real-world images captured in real-time.

The rest of the document is organized as follows.

Chapter II explains the available lab-controlled and in-the-wild data-sets used for my experiments. It will include the establishment of the baseline models used for comparing my approach's performance. The experiments conducted on the baseline models for comparison are also included. Chapter III gives a description of the proposed hybrid network. The experiments on the proposed model and the comparative analysis are included as well. The real-time demonstration of facial expression recognition is explained along with the conducted experiments in Chapter IV. I finally conclude in Chapter V.

CHAPTER 2

DATASETS AND BASELINE MODELS

As mentioned in the previous chapter, the hybrid model is tested on two in-the-wild datasets namely RAF-DB, AffectNet and one lab-controlled dataset, CK+. There are several problems associated with in-the-wild datasets which are not present in lab-controlled datasets and therefore there have been networks performing exceptionally well on lab-controlled datasets but fail to do so on in-the-wild datasets.

The convolutional neural networks trained on ImageNet and FaceNet are used as baselines. The results obtained from these baselines are compared with the proposed method. Initially, in this chapter, a description of the datasets is provided. In the latter part of this chapter, the baseline models used for comparison are explained. In the end, the experiments conducted on the baselines in order to compare with the proposed model are given.

2.1 DATA-SETS FOR FACIAL EXPRESSION RECOGNITION

Facial expression datasets are small and demand knowledge transfer to train a neural network. The domain of the source network plays a huge role in the performance of the target network. Closer the source domain is to the target domain, more is the accuracy of the model. Lab controlled datasets have a full-frontal face, almost no occlusions but it is not the case for in-the-wild datasets. They have varied head poses and occlusions.

Figure 2. Sample image from each class of RAF-DB dataset. The images have occlusions, varied head poses and lighting conditions.

2.1.1 RAF-DB Dataset:

RAF-DB is an in-the-wild dataset for facial expression recognition. RAF-DB has around 30,000 images with natural expressions downloaded from Flickr. These images were taken in the natural setting and labeled manually by several votes. A total of 40 annotators are used to label the images independently. These images include faces of people belonging to different ethnicity, age, and gender. This database has 5 accurate landmark locations along with 37 automatic landmark locations for each image. Such features make it a rich database. There are seven classes – Angry, Surprise, Happy, Sad, Neutral, Disgust and Fear. RAF-DB is split into train and validation set. The training set contains 12,771 images and the validation set contains 3068 images. Originally, these

images are of varied size. This is obvious as they are the images collected from the internet. I used the face cropping provided by the author and resized them to 224 x 224 so that they can be used as input to the VGG16 network. The RAF-DB dataset also suffers from high-class imbalance.

The class happy has the highest number of images and the class fear has the lowest. In order to handle the class imbalance, I utilized balanced mini-batches which contain an equal number of images from all the classes. For data augmentation, I used horizontal flips and random cropping on the training images. In horizontal flips, the image is flipped over the vertical axis and added to the training data. In random cropping technique, crops from the four corners and the center of the image are taken to form new images. These techniques are used to prevent overfitting during training and increase the robustness of the network. Due to the high imbalance in the class ratios, the metric used is the mean per class accuracy or the diagonal mean of the confusion matrix.

2.1.2 AffectNet Dataset:

AffectNet is the largest labeled facial expression database and has around 500,000 images downloaded from the internet using various search terms from different languages. The images are downloaded from three major search engines using more than 1250 emotion-related images. Six languages have been used to search these images. They provide valence and arousal values along with the expressions. They have eleven emotion and non-emotion categorical labels (Neutral, Happiness, Sadness, Surprise, Fear, Disgust, Anger, Contempt, None, Uncertain and No-Face). Around half of the total images were manually annotated in order to detect the presence of the emotion labels for categorical

model and valence and arousal values for the dimensional model. I used only the base seven categories as in RAF-DB dataset for the experiments. I used the face coordinates from the dataset and resized them to 224 x 224 for input to the VGG16 network.

Figure 3. Sample image from each class of AffectNet dataset. These images do not have uniform lighting conditions and full-frontal face in the image.

AffectNet is also split into train and validation set. The validation set has 500 images from each class for a total of 3500 images. Just like other in-the-wild datasets, AffectNet also suffers from class imbalance issue but the validation set has been designed to have balanced classes. Hence, accuracy and mean per class accuracy provide the same result and can be used to measure the performance of the model. I used the images of the

seven expressions from the first 200,000 samples of the dataset for a total of around 140,000 to train my model and then test the performance using the validation set.

2.1.2 CK+ Dataset:

CK+ is a facial expression dataset captured in the lab-controlled setting. It is the second version of the Cohn-Kanade AU-Coded Facial Expression Database which is made publicly available. This extended version contains posed, spontaneous expressions along with additional metadata. CK+ has 22% more sequences than the previous version for the posed expressions and 27% more number of subjects. This increases the diversity in the images. It has 593 video sequences from 123 subjects. All video sequences are grey-scaled and have a resolution of 640 x 490. CK+ dataset has a total of eight classes – Happy, Sad, Anger, Fear, Disgust, Neutral, Surprise and Contempt. FACS encoding is used for the target expression of each sequence. I used the first seven classes only excluding the Contempt class.

These video sequences start with a neutral expression and shift to the peak expression. I extract the last two frames from the video for the peak expression and the first frame for the neutral expression. I used Viola-Jones (V&J) face detector to detect the faces from the image and removed the background from the images.

Figure 4. Sample image from each class of CK+ dataset. As these are lab-controlled images, the lighting condition is uniform as well as there is no occlusion or variation in the head pose.

2.2 PROBLEMS WITH IN-THE-WILD DATASETS

The in-the-wild datasets are difficult to handle as they contain real-world images taken from the internet. Unlike lab-controlled images, they have a lot of variations in the images. The problems in such datasets are shown in Figure 4.

Figure 5. Problems encountered while using in-the-wild dataset. The images have varied illumination and head poses. Objects like book, glasses block the entire frontal face view and create an occlusion.

2.2.1 Occlusion:

One of the most common problems in in-the-wild datasets is occlusion. Any images that have objects like eyeglasses or hand on a face are known as occluded images. The objects creating occlusion need not be large. Even hair that falls on the forehead and eyes can create an occlusion. Such objects block the view of the frontal face creating difficulties in features extraction.

2.2.2 Head pose:

Another major problem with the in-the-wild dataset is a large variation in the head poses. As these images are generally taken from the internet, almost every image has a different head pose as shown in Fig. 4. Such images will have faces that are visible from a direction only. Extracting the face properly from such images is a major challenge. Various

algorithms have been implemented to overcome this problem and detect the face even if it is titled in any direction.

2.2.3 Illumination:

The lighting conditions in which the images are taken also affect the classification task immensely. In case of very bright or very low light, the faces are not totally clear. Also, there are images where you have different colors of light like green, red, blue, etc. Such images are shown in Figure 5. Such non-uniform lighting condition can block the view of a few parts of the face which can lead to the wrong prediction of the facial expression.

2.2.4 Class Imbalance:

As mentioned earlier, the facial expression recognition datasets are small. Apart from having a smaller number of total images, these datasets have the issue of class imbalance. This means that some classes have a very high number of images while some have a very small number of images. In such cases, the network can learn the features properly from the class containing the high number of images and learns poor features from the deficient class. There are various ways like oversampling, undersampling, to handle this issue. In oversampling, the data augmentation techniques like horizontal flipping, rotation are used to increase the training data of the classes which contain a smaller number of images. Undersampling is an opposite technique of the oversampling. In undersampling, the training images for classes with a larger number of images are reduced so that all the classes contain an almost equal number of images. One of the most popular ways is to use

balanced batches [36] where every batch will have an equal number of images from each class. Here, we do not increase or decrease the number of training images in the classes. As the batch has an equal number of images from all the classes, the network learns features from each class properly and there is no bias created for a particular class while the prediction phase.

2.3 BASELINE MODELS

For comparison of the proposed model, we use VGG-Net trained on ImageNet dataset and FaceNet dataset as baselines. These networks have single domain knowledge and therefore are compared with my approach in order to show that a combination of two domain knowledge is better than a piece of single domain knowledge.

2.3.1 ImageNet:

ImageNet database was first presented in Conference on Computer Vision and Pattern Recognition (CVPR) 2009 as a poster by Princeton University's researchers. It is a dataset containing 1000 classes. It is organized according to the WordNet hierarchy. Each node in this hierarchy represents a set of images. Every category in WordNet is represented by a set of closely-related words which are called as synonym set or synset. In ImageNet, there are around 1000 images provided for every synset.

This dataset contains over 14 million images and has proven to be one of the largest visual datasets. It contains a wide variety of classes ranging from different animals to different kinds of objects. The images in ImageNet are quality-controlled and manually annotated. This database uses crowdsourcing of annotations. For image-level annotations,

the presence of an object in the image is determined. For object-level annotations, the object is indicated with the use of a bounding box around it.

2.3.2 FaceNet:

FaceNet is a dataset containing images of faces of different celebrities from across the world. FacNet is quite like ImageNet dataset when it comes to size but the difference lies in the type of images. ImageNet contains images of a lot of categories like animals and objects. On the other hand, FaceNet contains images of human faces only. It has over 2 million images. As this dataset contains faces of celebrities from all over the world, it has no classes. Instead, it makes use of embeddings for each input vector. Using the K-Nearest Neighbor algorithm, the distance between the two images is found out. If it below a certain threshold, then they are images of the same celebrity.

2.3.3 VGG-16 Network:

VGG-16 is a convolutional neural network which gained popularity due to its excellent performance on ImageNet dataset. It outperforms the famous AlexNet by replacing the large sized filters with 3x3 filters. The architecture of VGG-16 network contains 16 layers which are shown in Figure 6.

The input is a 224x224 RGB image which goes through a stack of convolutional layers. These have a fixed kernel size of 3x3. In total, there are five max-pooling layers with a kernel size of 2x2. The padding is used in order to preserve the size after the convolution operation is performed. After the stack of convolutional and max-pooling layers with ReLU activation, there are two fully-connected (FC) layers. The first layer

contains 4096 nodes and the last one contains 1000 nodes. These two layers can be fine-tuned in order to use VGG-16 network for different datasets.

For my experiments, I use VGG-16 network with 2048 nodes in the first fully-connected layer and 1000 nodes in the last layer. I use two VGG-16 networks, one is trained on ImageNet dataset and the other on FaceNet dataset. I compare the performance of these networks on in-the-wild datasets with my approach.

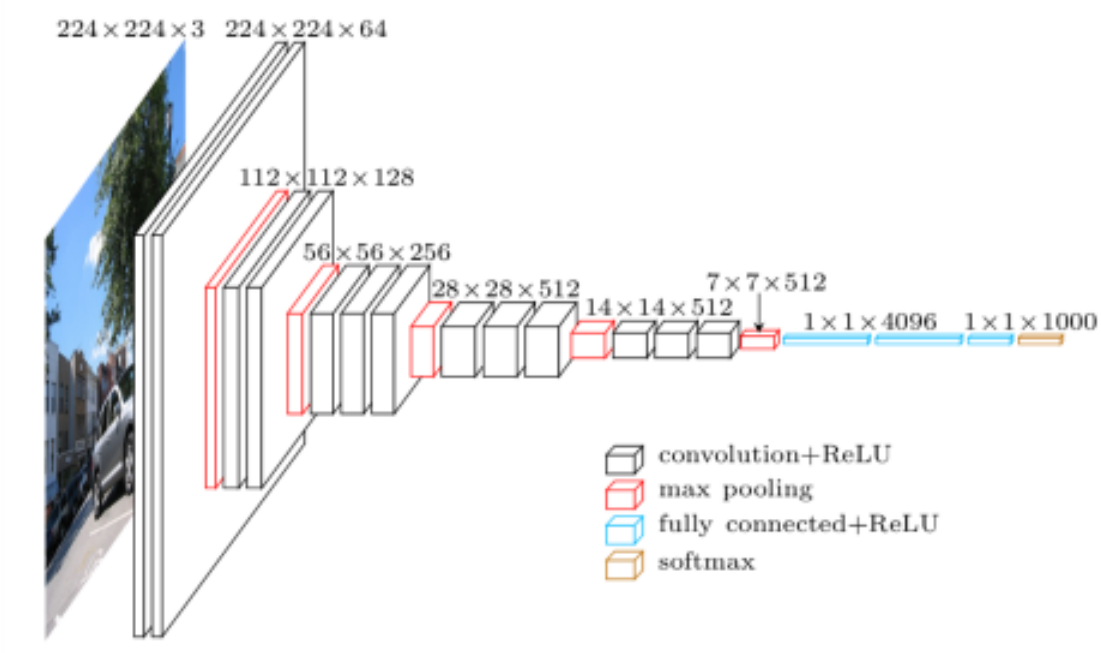

Figure 6. The architecture of VGG-16 network. The input is a 224x224 RGB image and the output is a class label out of 1000 classes.

2.4 METRIC USED FOR COMPARISON

Generally, accuracy on the test dataset is used for comparing the performance of various deep learning models. The accuracy is the ratio of images which have correctly

predicted labels to the total number of images. It is a very good measure when all the classes have almost the same number of images in it. Accuracy is not considered a good measure of performance when the dataset contains high-class imbalance like the in-the-wild datasets. For example, if there are 1000 testing images which have 900 images from happy class and 100 from all other classes. Even if all these 100 images are predicted wrong, the accuracy will be 90% which indicates that the model works quite well. But this is not true as the model works good only for the class with a large number of images and fails to perform well in other classes.

Therefore, I use a different performance metric to compare my approach with the baseline models. It is mean per class accuracy. It is the ratio of correctly predicted images from a class to the total number of images in that class. This indicates how the network performs for a class rather than the entire dataset.

CHAPTER 3

PROPOSED APPROACH – FUSION LAYER

This chapter provides the description of the proposed approach where the domain knowledge from two different networks is combined in a way that enhances the feature extraction. The knowledge from two domains not only leads to richer feature extraction but increases the prediction performance of the model. My approach is tested on two in-the-wild datasets and one lab-controlled dataset for cross-database study. The first part of this chapter will explain the approach and network architecture. The later part will list the experiments and results performed. It also contains the comparative analysis done with the baselines mentioned in the previous chapter.

3.1 HYBRID NEURAL NETWORK WITH FUSION LAYER

The facial expression recognition task demands the transfer of domain knowledge as the datasets are too small. The general approach for this task is to extract features from a network trained on ImageNet dataset. The ImageNet dataset contains around 14 million images of 1000 different classes. The network trained on ImageNet can extract good overall general features from an image, but the features are still far from the facial features. Therefore, performance is not exceptionally well. Recent methods use FaceNet dataset. FaceNet [20] is designed for the face recognition task but lacks the discriminating power of the facial expressions' characteristics. Still, the FaceNet domain is close to facial expression domain as compared to ImageNet. Therefore, the network trained on FaceNet can extract good facial features that can be used for the classification. This works well only

for the lab-controlled datasets because they have the full-frontal face, almost no occlusions which are not true for in-the-wild datasets.

To address this issue, I proposed a system that utilizes knowledge from both the domains to classify expressions better. FaceNet network extracts good facial features and ImageNet network extracts good generic features of the image. The combination of the features extracted from the two datasets can be fused together to classify wild expressions better.

3.2 MODEL ARCHITECTURE:

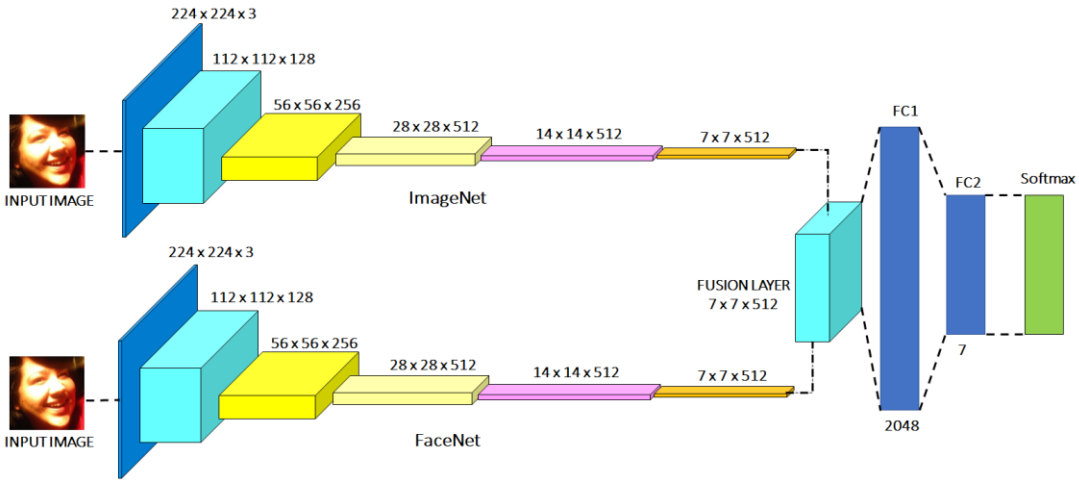

Figure 7: Proposed Model Architecture. Two VGG-16 networks trained on two different datasets are used to extract deep features which are merged using the Fusion Layer.

The model architecture is shown in Figure 7. It uses two VGG16 networks trained on ImageNet and FaceNet datasets. There are 13 convolutional layers and 5 max-pooling layers from the pre-trained VGG16 networks to extract the deep features of size 7 x 7 x

512. As fully connected layers are domain specific, I excluded them. Next, I introduce Fusion layer which merges the output features of the two networks. First, the features maps are mapped between the two networks with the most distinct distributions. To do so, I extract the features of all the samples from both the VGG16 networks. The mean value of all the samples is used to get the mean distribution of 7 x 7 x 512 features. Each channel can be considered a vector of 49 features and compared against different channels of the second vector using cosine similarity (Eq. 1). Each channel of the first VGG16 network is mapped to the most distinct channel of the second VGG16 network i.e. channels with the lowest cosine similarity value. These channels are arranged to put together like in Figure 8.

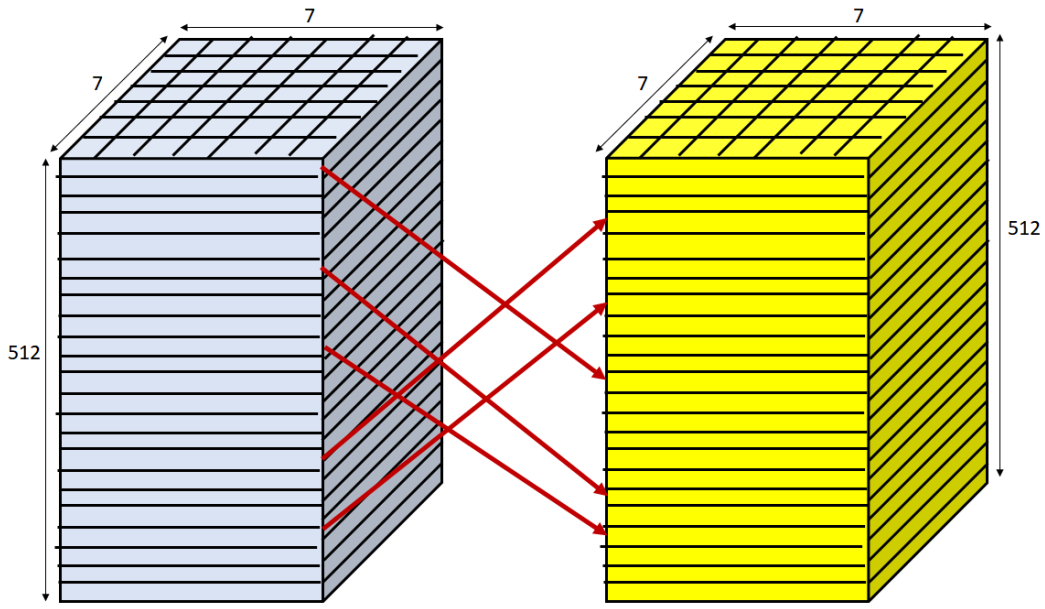

Figure 8: Channel Mapping. In the fusion layer, the channels from the output of both the networks are compared using cosine similarity and the most dissimilar channels are selected and arranged together.

$$Cos\theta = \frac{A \cdot B}{||A|| \cdot ||B||} = \frac{\sum_{i=1}^{d} A_i * B_i}{\sqrt{\sum_{i=1}^{d} A_i^2} \sqrt{\sum_{i=1}^{d} B_i^2}} \quad \text{.....(1)}$$

Next, we introduce the α alpha parameter which controls the weights of the channels and ranges between 0 and 1. First VGG network is multiplied by alpha € R^{512} and the second VGG network is multiplied by 1 - alpha. The channels are added together as per Eq. 2 to result in the original size of 7 x 7 x 512. The operation can be seen in Figure 9.

$$Output = \alpha \cdot feature_1 + (1 - \alpha) \cdot feature_2 \quad \text{.....(2)}$$

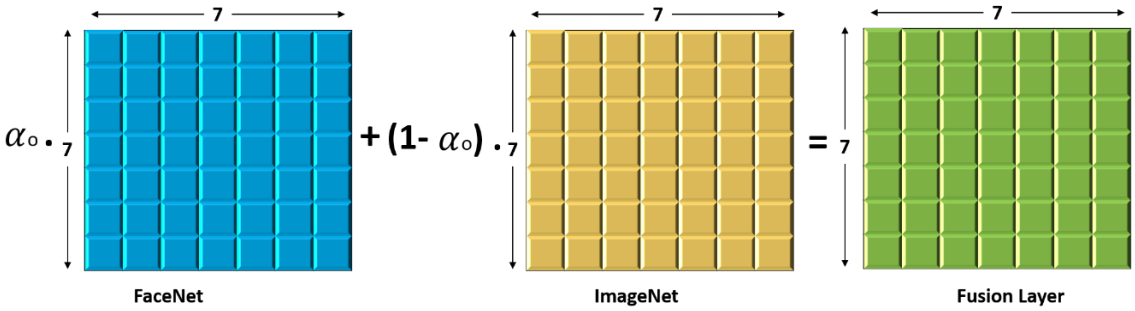

Figure 9: Fusion Layer. After selecting the most dissimilar channels, the first one is multiplied with a factor of alpha and the other with 1-alpha. They are then added to get the output of the Fusion Layer.

Using alpha, the network can emphasis which channel is better for the expression classification by assigning a higher weight to it. The alpha is initialized with a normal distribution having a mean of 0.5 and 0.1 standard deviation. The network backpropagates the error and learns the optimal value of the alpha. The extracted data is flattened and passed to the fully-connected layer of size 2048 neurons with dropout [23]. Dropout is

leaving some of the neurons in the hidden layers. These neurons which are chosen at random are not counted in the particular forward and backward pass. This is done in order to avoid overfitting of the model. The fully connected layer is further connected to the SoftMax activation. The SoftMax layer will give us the probability that the image belongs to each class calculated using equation 3.

$$P(\bar{y}) = \frac{e^{\bar{y}_i}}{\sum_j e^{\bar{y}_i}} \qquad \text{……..(3)}$$

The one with the highest probability is chosen as the predicted label. As these are probability values, they range between 0 and 1. The network is trained using the cross-entropy loss to increase the inter-separability of the classes given in equation 4.

$$Cross\ Entropy\ loss = -\sum_i^C y_i \log(\overline{y_i}) \qquad \text{……..(4)}$$

Fusion layer merges outputs of the two domains together across the channels. The channels of VGG-ImageNet that can classify expressions better gets more weight than the VGG-FaceNet and vice-versa. The combination of the two domains outperforms the individual domains.

3.3 EXPERIMENTS AND RESULTS

I used the Viola-Jones (V&J) [26] detector to detect the faces in images. This is a very popular algorithm because it is highly robust and works in real-time. Viola-Jones algorithm has four steps- Haar Feature Selection, Creating an integral image, AdaBoost training, and cascading classifiers. All the human faces share similar properties like the upper cheek's region is fairer as compared to the eyes region. Such properties are matched using Haar

Features. Next step is the creation of the integral image done in constant time. Therefore, they are fast as compared to any other algorithm. This is an image representation evaluating the rectangular features from the previous step. Next, the AdaBoost algorithm is used to choose the best features and use them to train the classifiers. At this stage of the algorithm, a strong classifier as a combination of several weak classifiers is obtained. The last step is to cascade these classifiers to get the final bounding box for the face in the image.

The network is trained on Nvidia Titan X GPU using a batch size of 64 for 8000 iterations with a learning rate of 0.0001 and exponential decay of 0.1 after 4000 steps. The RMS Optimizer is used to train the network with the momentum of 0.9. RMS Optimizer is used because it restricts the vertical direction oscillations. Therefore, a higher learning rate can be used to accelerate the process of convergence. The dropout rate of 0.5 is used and along with the weight decay of 0.0001 for regularization. All layers are initialized using Xavier initialization [9]. The layers need to be initialized such that the values are not too large or too small. In case they are too small, the signal will fade away eventually and won't be useful. In case they are too large, it will lead to the problem of exploding gradients. Xavier's initialization will make sure that the weights are initialized to values within a particular range, avoiding the problem of exploding and vanishing gradients.

3.3.1 Experiments on RAF-DB Dataset:

The RAF-DB dataset suffers from high-class imbalance and therefore balanced mini-batches which contain an equal number of images from all the classes is utilized. I used horizontal flips and random cropping on the training images for data augmentation.

	Anger	Disgust	Fear	Happy	Sad	Surprise	Neutral
Anger	**77.16**	6.79	1.23	5.56	1.23	4.94	3.09
Disgust	7.5	**55.63**	1.88	10	7.5	1.88	15.63
Fear	1.35	1.35	**64.86**	4.05	9.46	16.22	2.7
Happy	0.51	0.34	0.42	**91.73**	1.69	1.1	4.22
Sad	1.67	3.77	1.26	5.65	**77.62**	1.05	9
Surprise	1.22	1.82	1.82	3.95	3.04	**81.46**	6.69
Neutral	0.74	3.24	0.44	4.71	6.76	3.24	**80.88**

Table 1: Confusion Matrix for RAF-DB dataset. The rows represent the actual values and the columns represent the predicted values of the seven different labels. The diagonal elements represent the correct predictions.

The proposed model achieves 75.62% mean per class accuracy on RAF-DB dataset. The confusion matrix is displayed in Table 1 and the results are compared against other models in Table 2. As per the results, VGG-FaceNet performs better than VGG-ImageNet on facial expression dataset as it is closer to the target domain. Not only the Fusion layer shows a significant increase in the mean per class accuracy when compared to models with single domain knowledge, but it also exhibits an increase in accuracy across all classes.

	Anger	Disgust	Fear	Happy	Sad	Surprise	Neutral	Average
DLP-CNN	71.6	52.15	62.16	92.83	80.13	81.16	80.29	74.2
Deep Compact Model	74.47	67.57	46.88	82.28	57.95	84.57	59.12	67.55
MRE-CNN	83.95	57.5	60.81	88.78	79.92	86.02	80.15	76.73
VGG-ImageNet	64.81	37.5	48.65	86.16	67.15	74.77	69.85	64.12
VGG-FaceNet	74.69	55.63	60.81	90.89	77.2	79.94	79.26	74.05
Domain fusion (Ours)	**77.16**	**55.63**	**64.86**	**91.73**	**77.62**	**81.46**	**80.88**	**75.62**
Domain fusion with fixed $\alpha = 0.5$	79.63	53.75	59.46	90.55	77.41	81.76	82.21	74.97

Table 2: Comparison of different models on RAF-DB dataset. The rows represent different models used for comparison. The first 7 columns represent the class accuracy and the last column is the average accuracy over all the classes.

3.3.2 Experiments on AffectNet Dataset:

Only the base seven categories are used for the experiments. The face coordinates from the dataset are used and the images are resized to 224 x 224 for input to the VGG16 network. As the number of images per class is equal for the validation set, therefore the accuracy, as well as mean per class accuracy, yields the same result. Either of them can be used as a metric for comparison of the models. To maintain uniformity, I use class per mean accuracy as a measure of performance.

I used the images of the seven expressions from the first 200,000 samples of the dataset for a total of around 140,000 to train the model. The model achieves 53.6% accuracy on the validation set. I compared the fusion model against the individual domains and the results are present in Table 4. The confusion matrix for the fusion model is displayed in Table 3. From Table 4, it is clear that the domain fusion model outperforms the model with single domain knowledge. Not only the overall accuracy, but the mean per class accuracy is also better for the proposed model as compared to others.

	Anger	Disgust	Fear	Happy	Sad	Surprise	Neutral
Anger	**62.2**	5.6	2.4	5	10.6	4.4	9.8
Disgust	28.2	**29.8**	3.8	12	12.4	7.6	6.2
Fear	8	1.8	**41.8**	4.2	12	28.8	3.4
Happy	1.2	0.4	0.4	**91**	1.2	4.4	1.4
Sad	11.4	1.4	3	6.4	**60.8**	7	10
Surprise	4.2	2.6	7.6	12.8	7.6	**58**	7.2
Neutral	15.6	2.2	2.8	15.4	18.8	13	**32.2**

Table 3: Confusion table for AffectNet dataset. The rows in the table represent the actual values and the columns represent the predicted values. The diagonal elements of the table represent the correct predictions.

	Anger	Disgust	Fear	Happy	Sad	Surprise	Neutral	Average
VGG-ImageNet	55	22.2	35.4	89.2	49.6	51.2	28	47.22
VGG-FaceNet	60	28	37.4	89.8	59.4	59.2	35	52.68
Domain Fusion	**62.2**	**29.8**	**41.8**	**91**	**60.8**	**58**	**32.2**	**53.68**
Domain fusion with fixed $\alpha = 0.5$	51.4	15.4	27	92.8	44.2	33	78.6	48.91

Table 4: Comparison of domain fusion to individual domains on AffectNet dataset. The rows represent different models used for comparison. The first 7 columns represent the class accuracy and the last column is the average accuracy over all the classes.

3.3.3 Experiment on Alpha:

I performed an experiment on the α by keeping it at a constant value of 0.5 i.e. it is no longer a learnable parameter. This way, I could test whether the network is still able to learn the mapping. The results for RAD-DB are present in table 3 and for AffectNet in table 4. The network shows improvement for RAF-DB but not able to reach the performance of original architecture. In the case of AffectNet, the performance degrades even less than the VGG-FaceNet. Hence, it is important to keep α as the learnable parameter.

3.3.4 Experiments on CK+ Dataset:

After performing the experiments on two different in-the-wild datasets and demonstrating that the proposed approach provides comparable results on such datasets, I

performed experiments using CK+, a lab-controlled dataset. I tested the trained model on CK+ dataset to study the cross-database evaluation protocol on a lab-controlled dataset.

I used the models trained on RAF-DB and AffectNet to predict the expressions on the CK+ dataset. The RAF-DB model achieves an accuracy of 72.48% and AffectNet achieves 72.38% on CK+ dataset without any fine-tuning. The confusion matrix of RAF-DB is displayed in Table 5 and for AffectNet in Table 6. For both the models, many of the non-diagonal elements are zero which indicates that my proposed model can work on the lab-controlled datasets very well.

	Anger	Disgust	Fear	Happy	Sad	Surprise	Neutral
Anger	**3.33**	42.22	0	6.67	30	3.33	14.44
Disgust	10.17	**85.59**	0	3.39	0	0	0.85
Fear	0	4	**28**	8	40	16	4
Happy	0	0	0	**98.55**	1.45	0	0
Sad	0	12.5	0	0	**69.64**	7.14	10.71
Surprise	0	1.81	1.81	3.01	0.6	**89.76**	3.01
Neutral	0	8.26	0	13.46	9.17	4.28	**64.83**

Table 5: Confusion table for CK+ dataset using model trained on RAF-DB dataset. In the table, the rows represent the actual values and the columns represent the predicted values. The diagonal elements in the table represent the correct predictions.

	Anger	Disgust	Fear	Happy	Sad	Surprise	Neutral
Anger	**44.44**	8.88	0	8.88	15.55	0	22.22
Disgust	31.35	**66.94**	0	0	0	0	1.69
Fear	0	0	**30**	10	44	12	4
Happy	0	0	0	**100**	0	0	0
Sad	1.78	1.78	3.57	0	**83.92**	0	8.92
Surprise	0	0	9.03	0	1.2	**88.55**	1.2
Neutral	4.28	1.52	0	10.7	12.23	4.58	**66.66**

Table 6: Confusion table for CK+ dataset using model trained on AffectNet dataset. In the table, the rows represent the actual values and the columns represent the predicted values. The diagonal elements in the table represent the correct predictions.

3.4 ANALYSIS OF RESULTS

For experiments on RAF-DB dataset, this approach performs well on the Angry, Happy, Sad, Surprise and Neutral expressions. It performs average on the Disgust and Fear expressions. These two are the common in-the-wild expressions where the neural networks do not perform well. These are the classes which have the least number of training examples. One possible reason the neural networks do not perform well on these expressions can be lack of the training samples for these classes and highly varied distribution of expressions.

For the experiments performed on AffectNet database, there is a boost in the performance of Angry, Disgust, Fear, Happy and Sad expression classes in comparison to

FaceNet network. The results are similar to those of RAF-DB regarding the Disgust and Fear expression. Overall, for in-the-wild datasets, the FaceNet domain outperforms the ImageNet domain as it is closer to the target domain. The fusion of the two outperforms the individual ones.

For the cross-database study, there are two sets of results – one for the model trained on RAF-DB dataset and the other one trained on AffectNet dataset. RAF-DB model can classify Neutral, Surprise, Happy, Disgust and Sad expressions well but works poorly for Anger and Fear expressions. This indicates that the images for Anger and Fear expressions differ a lot in a lab controlled and the natural settings. Anger is mostly classified as Disgust and the resemblance can be seen in the sample images of the datasets. Just like the RAF-DB, AffectNet model works well on the Disgust, Happy, Sad, Surprise and Neutral and struggles on the Anger and Fear expressions further concreting the conclusion that natural fear and angry expressions differ a lot from the lab-controlled settings.

CHAPTER 4

IMPLEMENTING A REAL-TIME DEMO SYSTEM

This chapter provides a description of the real-time demonstration of facial expression recognition. First, the description of the hardware used is provided. The next part of the chapter contains the algorithms used for this live demonstration and the entire flow of it. The later part contains the results of the experiments conducted on the real-time demonstration. Here, I show how the model works in cases of different illumination, lighting, and occlusions just like the in-the-wild images. It also contains the analysis of the different algorithms I tried for this live demonstration. This real-time demonstration shows that the proposed model performs fast and well in real-time also.

4.1 PRACTICAL CONSIDERATIONS IN BUILDING REAL-TIME SYSTEMS

4.1.1. Real-time Video Streaming

In this real-time demonstration, I demonstrate the real-time facial expression recognition using the camera of iPhone 7 and the proposed model on the Lenovo Y50 laptop. Lenovo Y50 laptop has 1 MP (megapixel) camera. Initially, I used the webcam of Y50 in order to capture the frames, but the captured images are of low quality and dark. This can lead to the poor performance of the model. Therefore, we use the camera of the iPhone 7 cellphone to capture the images. It is a 12 MP camera which captures high-quality images which can enhance the performance of the model. The images generated are in either JPEG or HEIC format. The images are then transferred to a laptop and given as input to the model for predicting the facial expression.

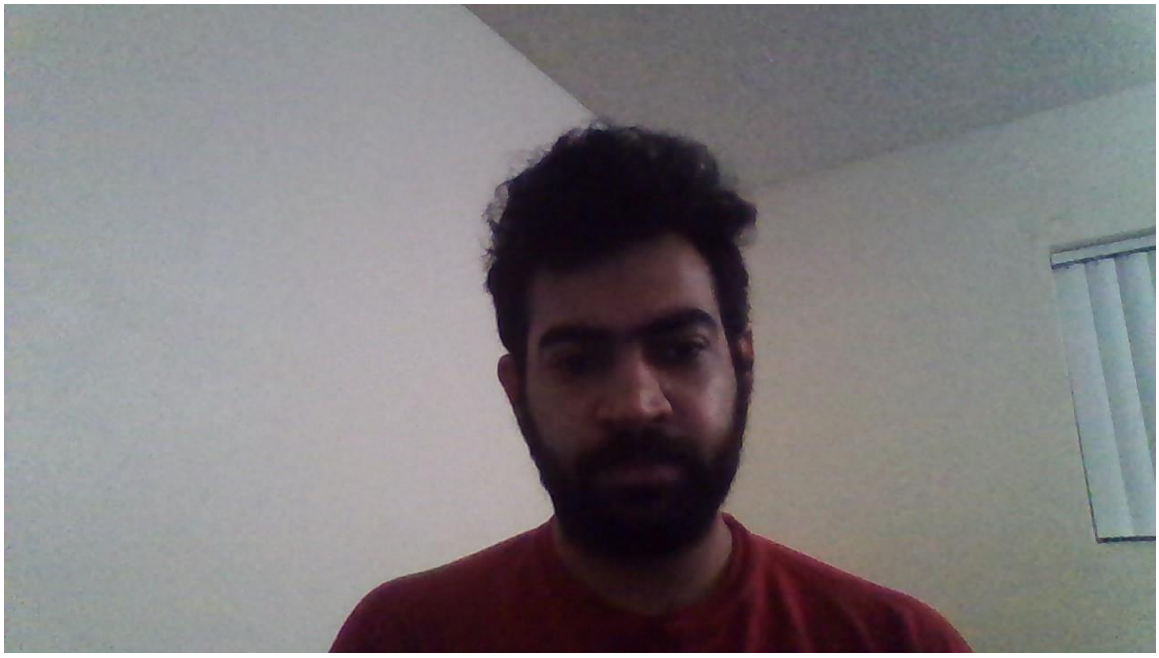

Figure 10: Capturing the frame from the real-time streaming video.

4.1.2. Face Detection

After the frames are captured from the real-time video using a webcam, the next step is to detect the face from the captured frames. I tried two algorithms for detecting the face. First, I used the Viola-Jones Algorithm which uses Haar features, but it fails in many real-time scenarios. For example, it does not detect a face if it is titled or there are any kind of occlusions. It detects the upright frontal faces. This is not desirable for real-time demonstration. So, I used the C++ D library shortly known as Dlib [29], to detect the faces in the captured frames.

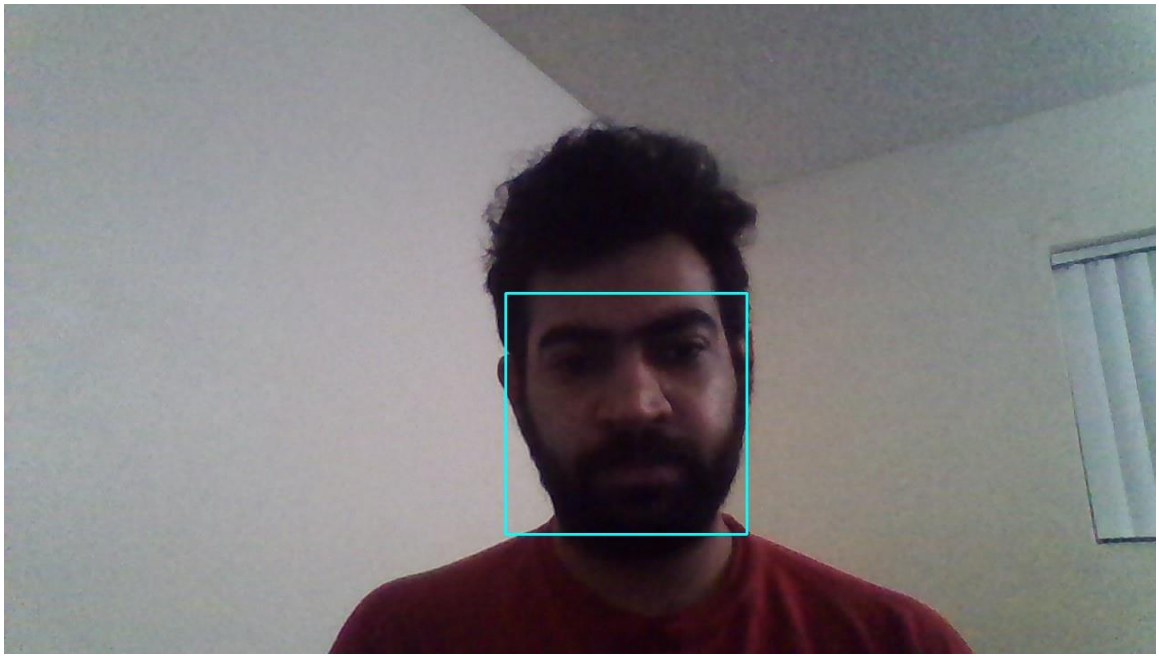

Figure 11: Detecting the face from the captured frames. HoG filter based Dlib is used for detecting the face.

Dlib is a toolkit that contains various machine learning algorithms to solve real-world problems. The face detector in Dlib is using an HoG filter. This face detector is based on five HoG filters namely front facing, right facing, left facing, front faced but rotated right and front faced but rotated left. Due to such filters, it can detect non-frontal faces as well as faces with occlusion. Figure 11 shows the bounding box showing the face detected. Also, it is the fastest method without using a GPU. Figure 12 shows some cases where Dlib outperforms Viola-Jones algorithm.

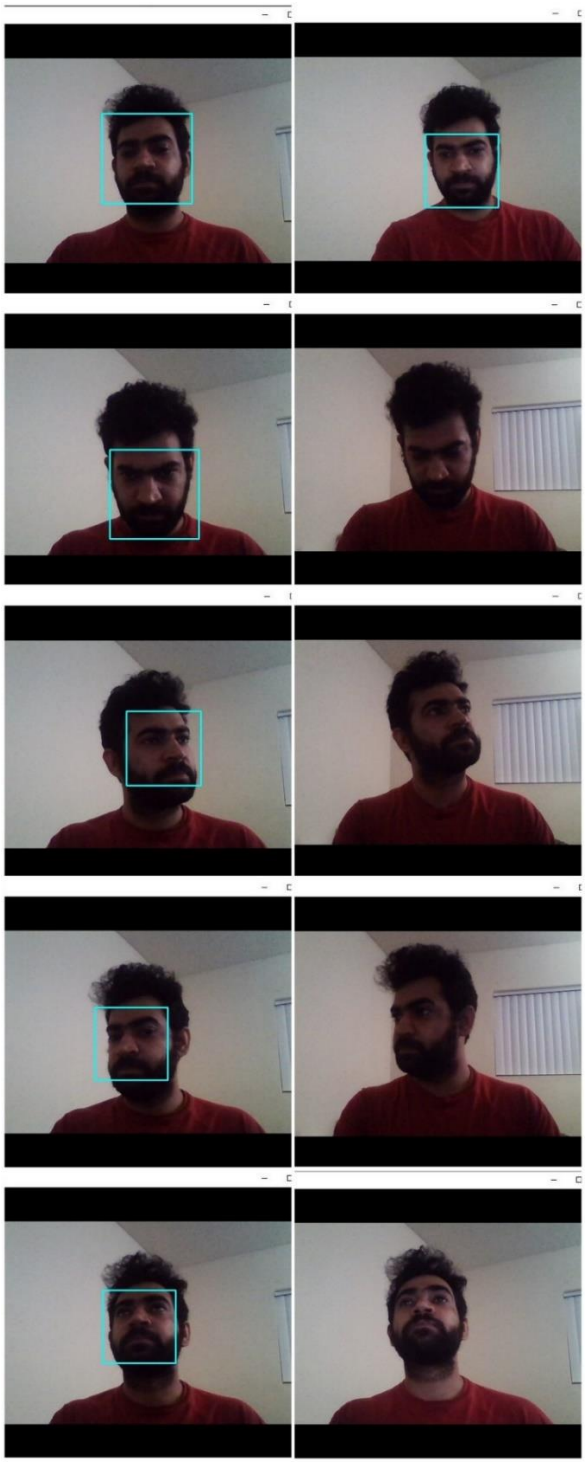

Figure 12: Comparison of Viola-Jones and Dlib Algorithm. The first column represents the face detection using Dlib method and the second column represents using the Viola-Jones algorithm.

In Figure 12, the Dlib performs well in real-time images than the Viola-Jones algorithm. Both algorithms work well for the upright frontal faces. In case of faces tilted to the sides, the Viola-Jones cannot produce a bounding box for the face in the image while Dlib library can detect it very accurately. The same result is seen when the faces are titled in either upward or downward direction. After detecting the face, it detects the 68 facial landmarks as well which will be used for face alignment which is explained in the next section. Figure 13 shows the facial landmarks given as output.

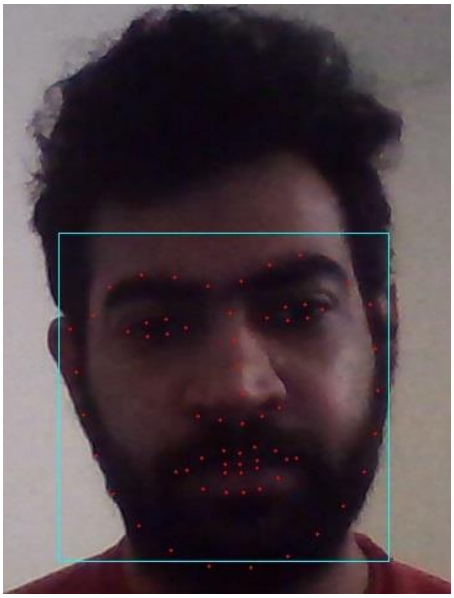

Figure 13: 68 facial landmarks given by the Dlib Algorithm

4.1.3. Face Alignment

After the facial landmarks are obtained, the next step is to align the face. Every eye has 6 facial landmarks as shown in Figure 13. To find the center of the eye, we take the mean point of all these 6 facial landmarks. Therefore, we get the two centers, one for each eye. Next, we find the height and width between these two points. The ratio of height to

the width will give the angle between the two points. In order to align them, the angle between them needs to be zero. This infers that the difference in the height needs to be zero. The calculated angle will help us in making the height zero. We need to rotate in the opposite direction of the calculated angle to make it zero. To obtain the axis over which the rotation needs to be done, I calculated the midpoint of the two centers of the eye. The vertical line passing through this center is used as the axis for rotation. Now, we can rotate the image in the required direction to align it. For example, if the calculated angle is 30º, then rotate it by 30º in the opposite direction over the given axis to get an aligned face.

Face alignment is not just about rotation, it is about the amount of face exposure as well. For example, the face is not considered as aligned even if it is too zoomed in or too zoomed out. For proper feature extraction, a proper amount of face exposure is necessary. To accommodate this problem, I fixed the distance of 0.35 time of the total distance from the boundaries of the image where the center of the left and right eye should lie. Also, I specified the distance between them to be 0.3. So, the image will be zoomed in and out till these facial landmarks are the specified places and I get the appropriate face exposure required for accurate prediction. The rotation of the face and the affine transformations were done using the OpenCV library.

Face Alignment is an important part because experiments have shown that the models which work on aligned faces as input have richer feature extraction. As a result, the accuracy of the model increases. With aligned faces, the feature extraction process becomes simpler.

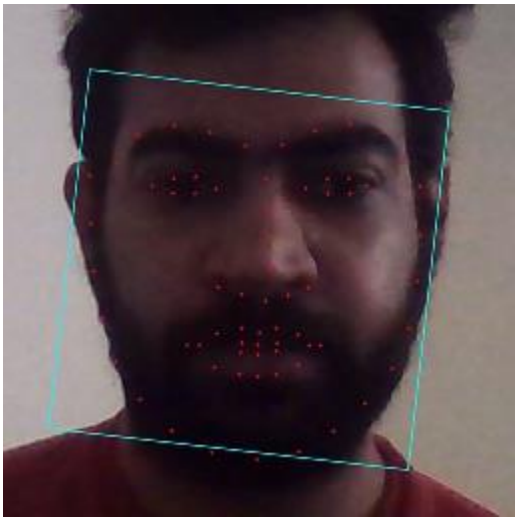

Figure 14: Using facial landmarks to perform face alignment step

4.1.4. Resizing

The output from the previous step needs to be resized in order to be given as input to the fusion layer model for predicting facial expression recognition. The image is resized to a size of 256x256. To further increase the facial exposure, I took the center crop of this resized image. The size of the resulting image is 224x224 x 3. This is the input image size required by my proposed model.

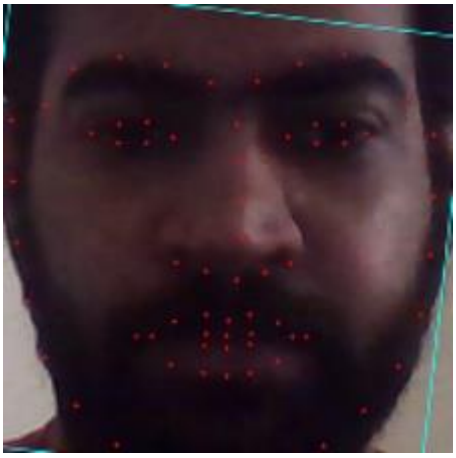

Figure 15: The aligned face is cropped and resized to get 224x224 size image.

4.1.5. Handling varying illumination conditions

After face alignment, different illumination conditions need to be handled. To overcome this problem, two techniques can be used – Min-max normalization [30] and Histogram Equalization [31]. In min-max normalization, all the input values are mapped between 0 to 1. The formula used is in equation 5:

$$y = (x - min)/(max - min) \qquad \text{……..(5)}$$

where x is the input, min is the minimum value in x and max is the maximum value in x. Another technique is known as Histogram Equalization. In this technique, the overall contrast of the image is increased. This is one of the most popular methods for dealing with illumination problems as it performs exceptionally well. It handles the varying illumination by spreading the frequently occurring pixel values. The first step is to calculate the probability density function of the input followed by the calculation of cumulative distribution function. Next, multiply the cumulative distributive function values with the respective pixel values. Lastly, the old pixel values should be mapped to the new values. Histogram equalization is originally done on the grayscale images. To apply it to color images, it should be applied separately to Red, Green and Blue channels. This changes the relative distribution of the image and therefore results in large dramatic changes. To avoid it, I converted it to YUV color space. Then, the histogram equalization is applied to the V channel. Figure 16 displays the image after histogram equalization. This image is given as input to the model which gives the corresponding facial expression as the output.

Figure 16: The final image after histogram equalization which will be fed to the model as input.

4.1.6 Predictions

The processed image in input to the network and the output are the probabilities of the seven classes. The class with the highest probability is predicted as the label.

4.2 SAMPLE DEMO IMPLEMENTATIONS

I used the Lenovo Y50 laptop to conduct the real-time facial expression recognition demo. For every frame the camera captures, the face is detected using Dlib algorithm, but the prediction is made after every 30 frames. It was observed that when the experiments are carried out on the pictures captured by the webcam of the laptop, the predicted labels are not very accurate. The experimental setup for the real-time demonstration is shown in Figure 17. The webcam captures the images. Pre-processing techniques are applied to it. The model gets these images as input and the output is displayed on the command prompt. The interface of this demonstration is shown in Figure 18. The window on the left shows the output of the image captured by the webcam. The real-time images are shown in the

right window. In Figure 18, initially, the output is neutral. Once the expression changes to surprise, the label surprise is displayed on the command prompt.

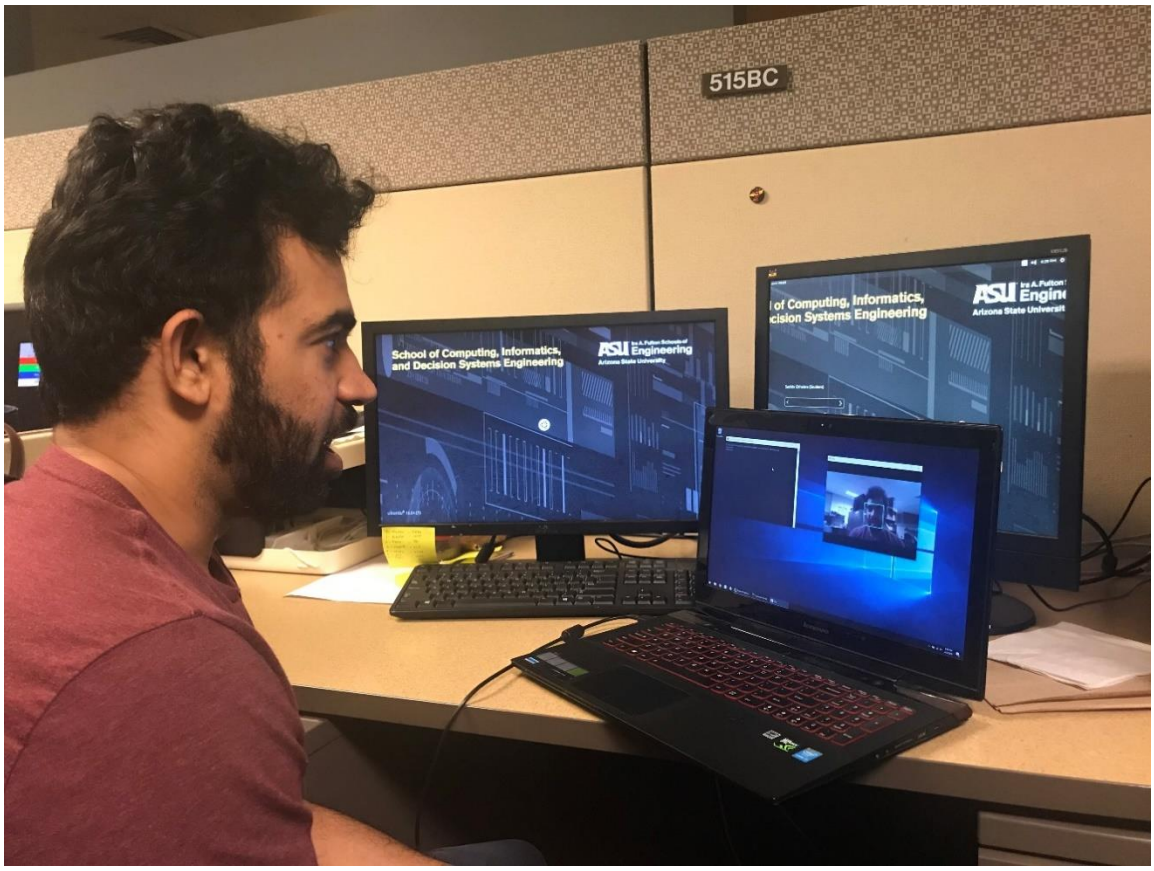

Figure 17: Experimental setup of real-time facial expression recognition system using the webcam of the laptop.

In some cases, the model predicts the expressions correctly but fails in some cases. For example, the surprise expression is quite distinguishable from other expressions and can be predicted by the model for the pictures captured by the webcam, but it does not make good predictions on expressions like anger and fear.

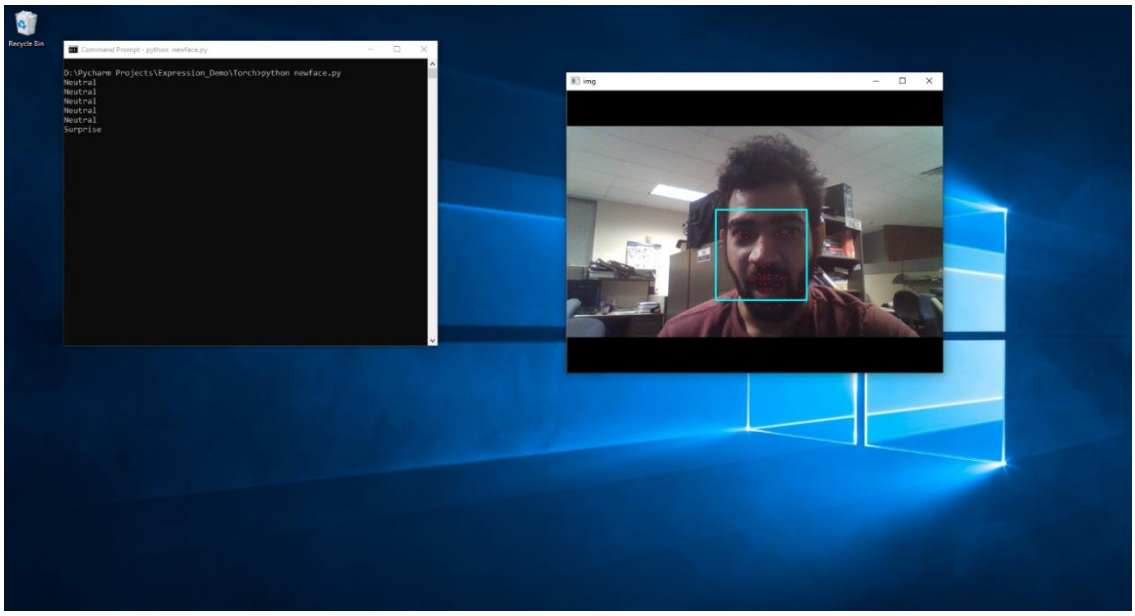

Figure 18: Example of facial expression real-time demonstration. The image captured and the corresponding label is shown.

As the model performs well on in-the-wild datasets, the model working poorly on the webcam images can be a problem of the quality of the images. As the webcam has a resolution of 1 MP only, the quality of the images is quite poor. extracted are not good enough to make a correct prediction.

To further test it, I captured the images using the iPhone 7 camera which has a resolution of 12 MP. The images captured by this camera are better than ones captured by the webcam. I present the various experiments I conducted to show the accurate results of the proposed model on the images captured in real-time using the iPhone 7 camera. The images taken using the iPhone camera are transferred to the laptop. Then, it is passed to the model for recognizing the facial expression.

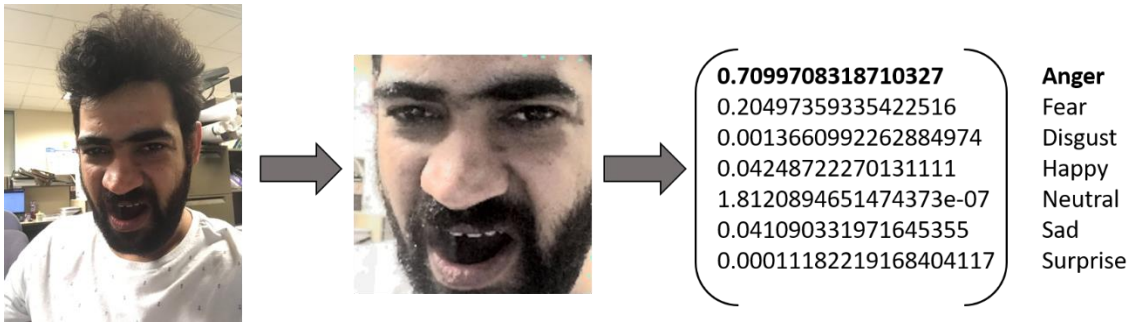

Figure 19: The output produced by the model for images captured using the iPhone camera for angry expression.

In Figure 19, the first image is the one captured using the iPhone camera. Applying the techniques mentioned in the previous section, the input for the model is obtained. After passing it to the model, the probabilities that the image belongs to a class are obtained. Here, the highest is of Anger which is the true label as well.

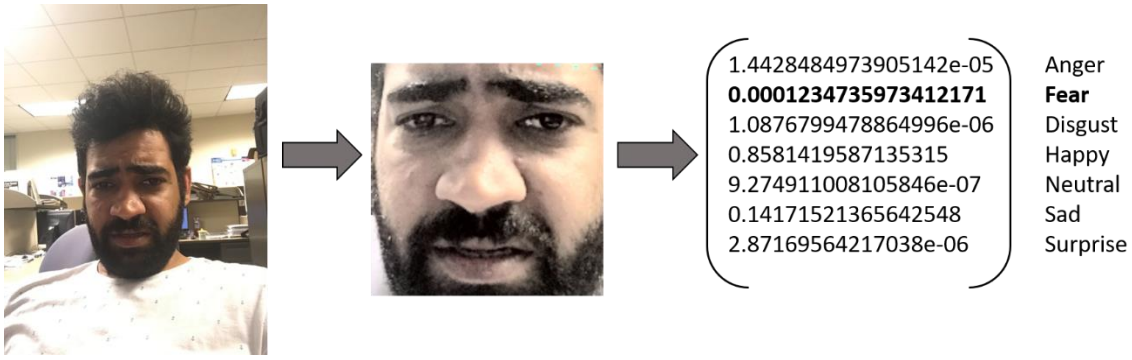

Figure 20: The output produced by the model for images captured using the iPhone camera for fear expression.

Figure 20 shows an image of fear expression. Here, though the correct label is chosen, we can see that the probabilities are quite low. Fear is one of the classes with very a smaller number of images. So, the probability values are too low.

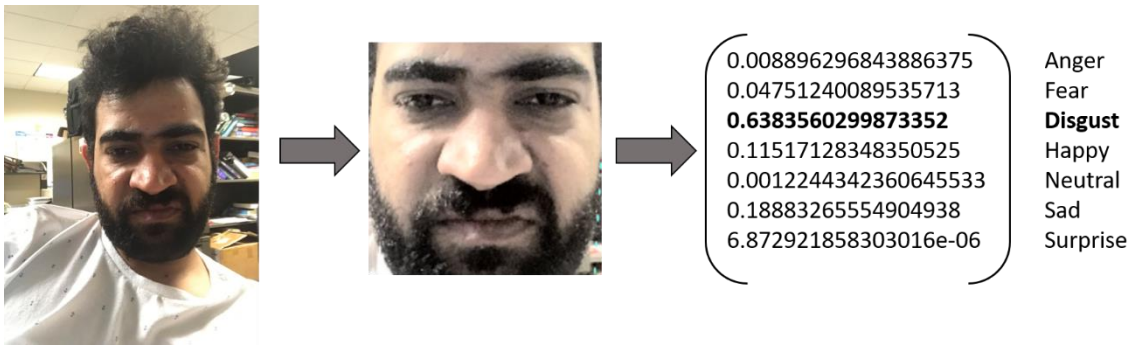

Figure 21: The output produced by the model for images captured using the iPhone camera for disgust expression.

Disgust is one of the classes which has a fewer number of images as compared to classes like Happy and Neutral. But it has a greater number of images than class Fear. From Figure 21, it is clear that the model performs well on Disgust. It has the highest probability and no other class is even closer to that probability value. This indicates that the network is quite confident about the prediction.

Figure 22: The output produced by the model for images captured using the iPhone camera for happy expression.

Happy is the class with the highest number of images. It is much easier to get images with happy faces than any other expression. Figure 22 shows that the predicted probability for class Happy is 1, which indicates that the network is 100% sure that it is an

image belonging to Happy class. The network can be so confident about the decision because it has learned a lot of features for this class as a result of more training data. The model performs best on Happy class.

Figure 23: The output produced by the model for images captured using the iPhone camera for a neutral expression.

The neutral class has a moderate number of images. These images are enough for the model to learn good features and output a high mean per class accuracy. Figure 23 shows that the model predicts the neutral expression with almost 0.75 probability, with no other class probability even closer to it.

Figure 24: The output produced by the model for images captured using the iPhone camera for sad expression.

According to the output class probabilities shown in Figure 24 for an input image belonging to Sad class, the model performs exceptionally well for this class. The size of the training set for this class is enough for the network to learn generalized features. It predicts the class label as sad with 99% confidence.

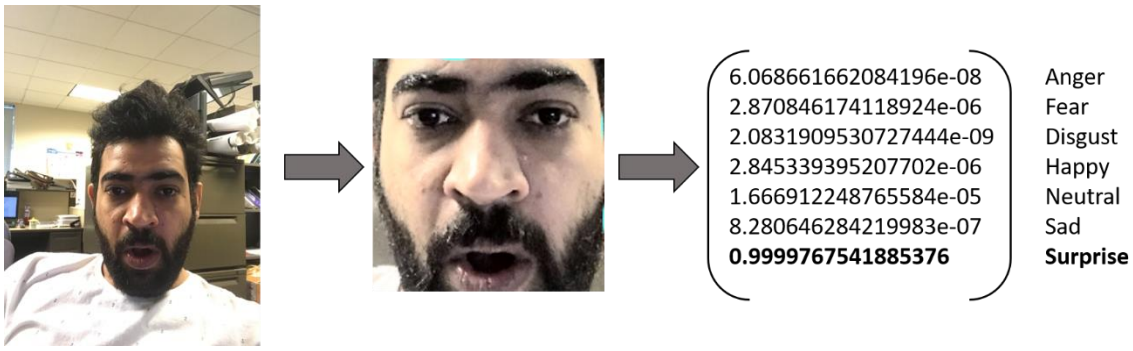

Figure 25: The output produced by the model for images captured using the iPhone camera for Surprise expression.

Similar to the sad class, the model gives highly accurate results for the images belonging to the surprise class. As shown in Figure 25, the model can predict the given image with almost 99.99% confidence.

From the above results shown on the iPhone 7 camera images, it is clear that the model worked poorly on the webcam images as the quality of the images was quite low. For the images from iPhone 7 camera, the model could predict the correct expressions with high confidence. Therefore, the quality of the images needs to be good for enhanced feature extraction.

CHAPTER 5

CONCLUSION

Several methods have been proposed to detect the facial expressions in the images. Most of the methods use the lab-controlled data-sets which have controlled conditions. The lighting is uniform, and the images have an entire frontal face. The lab-controlled images have no occlusions in most of the cases. So, face detection becomes easier in such datasets. Further, the feature extraction process gets easier. Therefore, facial expression recognition on such datasets becomes a lot simpler than for in-the-wild datasets. For the latter type of datasets, the images are taken from the internet and are real-world images. Therefore, they have problems like a difference in lighting conditions, varying head poses, and a variety of occlusions like sunglasses, hands, hair, etc.

The main contributions of this work are listed below. The proposed hybrid network with fusion layer takes two networks pre-trained on different domains as input and combines the features to form a superior network. The experiments showcased that the hybrid network outperforms the individual networks. This hybrid network is the first that utilizes features from two different domains on the facial expression recognition problem. It achieves comparable results with the state-of-the-art methods on the RAF-DB and the AffectNet datasets. The quality of the features learned by the model is further tested by performing a cross-database study on the lab-controlled dataset, CK+. The model gives good results on the lab-controlled dataset as well. This shows that the model can work for both kinds of facial expression datasets- lab-controlled and in-the-wild datasets, unlike the

models which work exceptionally well on the lab-controlled datasets but fail to do so when it comes to in-the-wild datasets.

To showcase the speed and accurate behavior of the proposed model for real-time images, I conducted a real-time demonstration of the facial expression recognition model. The demonstration is done using the Lenovo Y50 laptop which has 1 MP resolution camera. Initially, the images from the webcam of the Lenovo Y50 laptop were used for the real-time demonstration. The model works well for some expressions while performed poorly on some expressions. I concluded that it did not work well for these images because the quality of the images was poor as the resolution of the webcam is 1 MP only. To show that the quality of the images is the cause the model performs poorly, I took the images from the iPhone 7 camera instead of the laptop webcam for the demonstration. As the resolution of the iPhone camera is 12 MP, much higher than the webcam, the quality of the images is enhanced. As a result, the model is able to predict the correct expression with high confidence. This concreted my conclusion that the quality of the images needs to be good enough for better feature extraction and high performance of the model. Also, I identified the challenges for the real-time demonstration. The real-time demonstration needs a camera with good resolution and a model performing exceptionally well on the dataset. The environmental conditions also play an important role in the demonstration. I also observed that the accuracy of the model decreased if the face alignment step is not done. The performance of the model gets boosted when the face alignment step is done as the features extracted are good and easily extracted.

Overall, this work was an attempt to combine domain knowledge from two different domains in order to enhance the performance on the facial expression recognition task. The

experimental results indicate that this method has outperformed the models using single domain knowledge. The real-time demonstration was an attempt to show that the proposed model performs well for images captured in real-time as well. It also showed that the proposed model can predict accurate expressions with high confidence and almost instantly. Therefore, the processing time for the proposed model is low. For future work, I plan to analyze the drawbacks of the fusion layer, work to eliminate them and enhance the performance of the model. I also plan to use state-of-the-art techniques like deep locality preserving loss and island loss to further increase the performance of the proposed network on in-the-wild datasets.

REFERENCES

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In International Conference on Computer Vision & Pattern Recognition (CVPR'05), Volume 1, pages 886–893. IEEE Computer Society, 2005.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei. Imagenet: A large-scale Hierarchical Image Database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[3] A. Dhall, R. Goecke, S. Lucey, T. Gedeon, et al. Collecting large, richly annotated facial-expression databases from movies. IEEE multimedia, 19(3):34–41, 2012.

[4] H. Ding, S. K. Zhou, and R. Chellappa. Facenet2expnet: Regularizing A Deep Face Recognition Net for Expression Recognition. In 2017, 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), pages 118–126. IEEE, 2017.

[5] P. Ekman. Facial expression and emotion. American psychologist, 48(4):384, 1993

[6] Y. Fan, J. C. Lam, and V. O. Li. Multi-region ensemble convolutional neural network for facial expression recognition. In International Conference on Artificial Neural Networks, pages 84–94. Springer, 2018.

[7] B. Fasel and J. Luettin. Automatic facial expression analysis: a survey. Pattern recognition, 36(1):259–275, 2003.

[8] M.-I. Georgescu, R. T. Ionescu, and M. Popescu. Local learning with deep and handcrafted features for facial expression recognition. arXiv preprint arXiv:1804.10892, 2018.

[9] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, pages 249–256, 2010.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[11] C.-M. Kuo, S.-H. Lai, and M. Sarkis. A compact deep learning model for robust facial expression recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 2121–2129, 2018.

[12] S. Li, W. Deng, and J. Du. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2852–2861, 2017.

[13] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. IEEE Transactions on Image processing, 11(4):467–476, 2002.

[14] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, pages 94–101. IEEE, 2010.

[15] M. J. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, and J. Budynek. The japanese female facial expression (jaffe) database. In Proceedings of third international conference on automatic face and gesture recognition, pages 14–16, 1998.

[16] A. Mollahosseini, B. Hasani, and M. H. Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. arXiv preprint arXiv:1708.03985, 2017.

[17] S. Neeta and S. Bhatia. Facial expression recognition. International Journal on Computer Science and Engineering, 2, 08 2010.

[18] 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 international conference on multimodal interaction, pages 443–449. ACM, 2015.

[19] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In bmvc, volume 1, page 6, 2015.

[20] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 815–823, 2015.

[21] C. Shan, S. Gong, and P. W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. Image and vision Computing, 27(6):803–816, 2009.

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1):1929–1958, 2014.

[24] Y. Tang. Challenges in representation learning: Facial expression recognition challenge implementation. University of Toronto, 2013.

[25] Y. Tang. Deep learning using linear support vector machines. arXiv preprint arXiv:1306.0239, 2013.

[26] P. Viola and M. J. Jones. Robust real-time face detection. International journal of computer vision, 57(2):137–154, 2004.

[27] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In Advances in neural information processing systems, pages 3320–3328, 2014.

[28] Z. Yu and C. Zhang. Image based static facial expression recognition with multiple deep network learning. In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, pages 435–442. ACM, 2015.

[29] King DE. Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research. 2009; 10(Jul):1755-8.

[30] Patro S, Sahu KK. Normalization: A preprocessing stage. arXiv preprint arXiv:1503.06462. 2015 Mar 19.

[31] Shin M, Kim M, Kwon DS. Baseline CNN structure analysis for facial expression recognition. In2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN) 2016 Aug 26 (pp. 724-729). IEEE.

[32] Karali A, Bassiouny A, El-Saban M. Facial expression recognition in the wild using rich deep features. In2015 IEEE International Conference on Image Processing (ICIP) 2015 Sep 27 (pp. 3442-3446). IEEE.

[33] Mayya V, Pai RM, Pai MM. Automatic facial expression recognition using DCNN. Procedia Computer Science. 2016 Jan 1;93:453-61.

[34] Ng HW, Nguyen VD, Vonikakis V, Winkler S. Deep learning for emotion recognition on small datasets using transfer learning. In Proceedings of the 2015 ACM on international conference on multimodal interaction 2015 Nov 9 (pp. 443-449). ACM.

[35] Ravi A. Pre-Trained Convolutional Neural Network Features for Facial Expression Recognition. arXiv preprint arXiv:1812.06387. 2018 Dec 16.

[36] Sönmez EB, Cangelosi A. Convolutional neural networks with balanced batches for facial expressions recognition. InNinth International Conference on Machine Vision (ICMV 2016) 2017 Mar 17 (Vol. 10341, p. 103410J). International Society for Optics and Photonics.

[37] Yang F, Zhang Q, Zheng C, Qiu G. In-the-wild facial expression recognition in extreme poses. In Ninth International Conference on Graphic and Image Processing (ICGIP 2017) 2018 Apr 10 (Vol. 10615, p. 106150P). International Society for Optics and Photonics.