Comparative Analysis of Facial Affect Detection Algorithms

Ashin Marin Thomas

Report submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Engineering
in
Electrical and Computer Engineering

Lynn Abbott, Co-chair
Myounghoon Jeon, Co-chair
Edward Fox

May 22, 2020
Blacksburg, Virginia

Acknowledgments

The idea for this project originated from the Nervtech driving simulator situated in the Mind Music Machine lab and Fetch robot in the Terrestrial Robotics Lab at Virginia Tech. I am grateful to all the committee members for their input and guidance, especially Dr. Myounghoon Jeon for his advice and motivation during each stage of the project. The paper named 'Facial Expression Recognition for Children: Can Existing Methods Tuned for Adults Be Adopted for Children?' [36] acts as the baseline for this project. I'm also thankful to the authors of the paper for collecting the datasets namely CAFE, CohnKanade+, OhioCompound Facial Expressions of Emotion and Multi-PIE.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Humans have the ability to detect and recognize a face among different faces. These days computers are also able to do the same. Face detection applications are used in smartphones for face lock and even for payments by physical cards, thereby increasing security for money transactions. The security has be extended in identifying criminals in data breaches. Expressions are the changes occurring on the human face to indicate a person's intent or emotional states. The resurgence of neural networks has improved several object detection algorithms, especially the face detection and facial expression detection. The artificial intelligence (AI) system can detect emotions by learning what each expression means and applying the AI algorithm to a new set of facial expressions. Emotional AI is a technology that is capable of reading, imitating, interpreting, and responding to human facial expressions and emotions. [2]

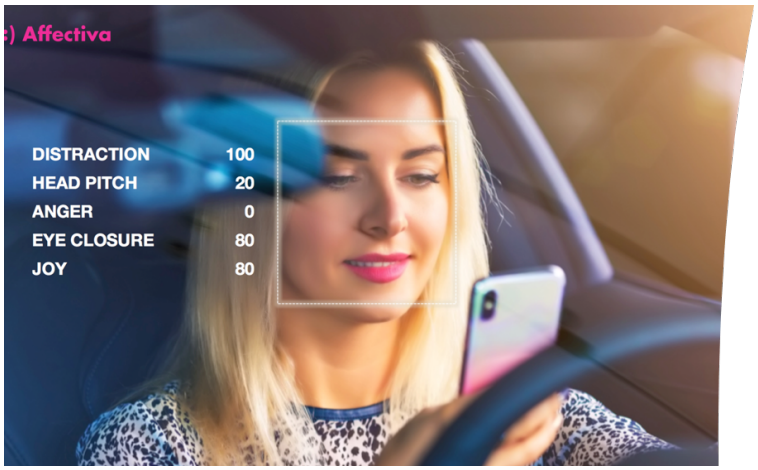

Figure 1.1: The Automotive AI system can recognize seven emotional metrics and as many as 20 facial expression metrics in drivers and passengers. [35]

.

Emotions can influence negatively on driving [14], [13]. A recent study also has shown the influence of emotions like anger on takeover performance in semi-autonomous vehicles [31]. Surprising things can happen in the car that might increase the risk of an accident. Thinking about the past, incidents that happen inside the car such as a disturbing phone message or spilling of the coffee or a car accident by the side of the road, or an animal injured by a car can can distract the driver and make him/her upset. The emotion detection and mitigation system in semi-automated vehicles detect driver drunkenness, stress, confusion, distraction or sleepiness. The vehicles can use AI algorithms to understand the human behavior context by taking real-time data from cameras in order to support the driver. The system can takeover in the autonomous driving mode to allow driver to recover [7].

To design an optimal emotion affect detection system in the automated vehicle context, the present project deals with comparing the current available facial expression detection algorithms, implement a new system and come up with a quantitative analysis for them.

1.1 Problem Statement

It is difficult to identify faces due to change in illumination, occlusion or noise in uncontrolled environment. This project tries to develop and then finalize an algorithm that can be used for real-time face emotion detection system. This system can be integrated in the Nervtech driving simulator which is located in the Mind-Music Machine Lab as shown in Figure 1.2.

Figure 1.2: Nervtech driving simulator.

.

The driving simulator has the following properties:

- Motion-based driving simulator
- 5.1 Surround Speakers
- Racing car seat, Steering wheel, pedals
- 120° horizontal field of view and consists

- Three 48-inch HD TVs

While most facial detection algorithms uses FER2013 (Facial Expression Recognition 2013) dataset [27],[29],[21], not much has been researched upon algorithms using other datasets. Each of the datasets mentioned in section 1.2 varies in size of dataset and in the dimension of each image in the dataset. Also, there has not been much study about analysing different emotion detection algorithms. In this study I focused on implementing the facial emotion detection algorithms for various datasets and give a comparative analysis of performance using the same datasets. The algorithms which have been implemented in the project are a CNN-based expression detection in Tensorflow, FaceNet using Transfer Learning and Capsule network. Each of these algorithms has been trained using various datasets to get the predicted results.

1.2 Datasets

The project deals with three facial expression datasets which are described as follows. These datasets have been chosen because the base paper [36] for the project uses these datasets to come up with the predicted results. Also, not much development has been done for developing or validating the emotion system using these datasets. Also, the datasets are well managed ones with emotion labels mentioned in separate files.

1. FER2013

   The data consists of 48*48 pixel grayscale images of faces. There are seven categories of images and each facial expression falls in these categories which are neutral+ six basic emotions (angry, disgust, fear, happiness, sadness and surprise) [6]. Each emotion is labelled from 0 to 6 sequentially. The train.csv file contains 2 columns, “emotion”

and “pixels”. The emotion column contains the numeric code for emotion in which the image lies into. The pixels column contains the string separated pixel values of each image. The dataset contains 35887 images. Emotion labels in the dataset are classfied as follows:

- 0: 4593 images *Angry*
- 1: 547 images *Disgust*
- 2: 5121 images *Fear*
- 3: 8989 images *Happy*
- 4: 6077 images *Sad*
- 5: 4002 images *Surprise*
- 6: 6198 images *Neutral*

FER2013 is the base dataset for the methods implemented in section 3, as these algorithms are already implemented or constructed using the FER dataset. The dataset is available in the following link [1].

2. Extended Cohn-Kanade (CK+)

   The CK+ database has number of sequences increased by 22% and the number of subjects by 27% from the original Cohn-Kanade database [18]. The target expression for each sequence is fully FACS (Facial Action Coding System) coded and emotion labels have been validated [18]. The CK+ comprises a total of 593 sequences across 123 subjects. Sequences range from neutral to peak expression as shown in Figure 1.3. For the project, I have taken 981 total images. Emotion labels in the dataset are classfied as follows:

   - 0: 135 images *Angry*

- 1: 177 images *Disgust*
- 2: 75 images *Fear*
- 3: 207 images *Happy*
- 4: 84 images *Sad*
- 5: 249 images *Surprise*
- 6: 54 images *Contempt*

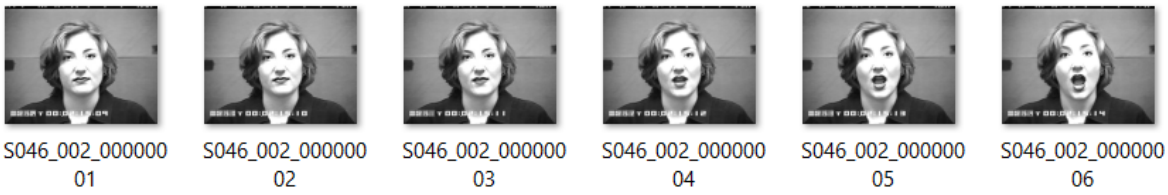

Figure 1.3: CK+ Image Sequence

.

The dataset is available at the following link [18].

3. Ohio Compound Facial Expressions

   The dataset consists of images with compound emotion categories. Compound emotions are those that can be constructed by combining basic component categories to create new ones. The basic component categories consist of categories such as happiness, surprise, anger, sadness, fear, and disgust. The dataset has 21 distinct emotion categories as shown in Figure 1.4. A Facial Action Coding System analysis shows that the production of these 21 categories is different but consistent with the subordinate categories they represent (e.g., a happily surprised expression combines muscle movements observed in happiness and surprise). Total images with pure emotions are 1610. Emotion labels in the dataset are classfied as follows:

- 0: 230 images *Angry*
- 1: 230 images *Disgust*
- 2: 230 images *Fear*
- 3: 230 images *Happy*
- 4: 230 images *Sad*
- 5: 230 images *Surprise*
- 6: 230 images *Neutral*

The database is obtained at the following link [4].

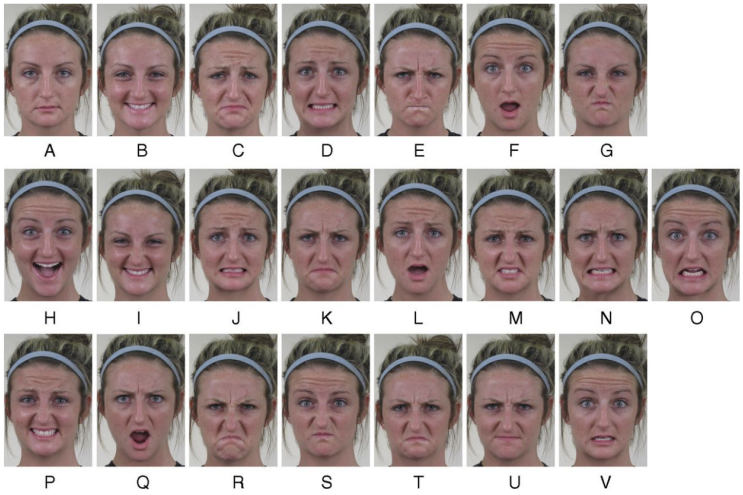

Figure 1.4: Sample images of the 22 categories in the database:(A) neutral, (B) happy, (C) sad, (D) fearful, (E) angry, (F) surprised, (G) disgusted, (H) happily surprised, (I) happily disgusted, (J) sadly fearful, (K) sadly angry, (L) sadly surprised, (M) sadly disgusted, (N) fearfully angry, (O) fearfully surprised, (P) fearfully disgusted, (Q) angrily surprised, (R) angrily disgusted, (S) disgustedly surprised, (T) appalled, (U) hatred, and (V) awed [33]

.

Chapter 2

Literature Review

The work by Zhi et al. in [36] investigates and finds a unique solution for the children expression detection problem. The authors of the paper used a mix of the publicly accessible datasets to create adults and children database. Most of the recognition methods used for children is made and tuned up using the adults face datasets. The paper deals this problem with SVM (Support Vector Machine) classification-based recognition. The paper examined the difference in facial expressions between children and adults and whether the SVM classifiers trained on one group of children dataset work on other groups. There are seven facial expressions that have been focused upon which are neutral, anger, disgust, fear, happiness, sadness and surprise. The constrained local neural fields method is used for facial feature extraction. To handle the different landmark distributions for different persons, normalization is used before feature extraction. As shown in Figure 2.1, for the left image the axes show the landmarks' position (number of pixels) on the original image, while the right image shows the landmarks of the same face after normalization.

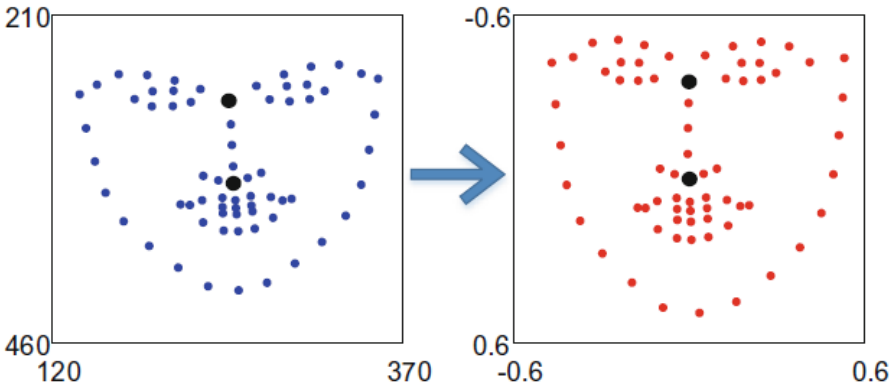

Figure 2.1: Data Normalization [36]

The below Figure 2.2, shows the differences for each emotion for an adult versus a child. As illustrated, children made larger mouth movements than adults when showing the emotions.

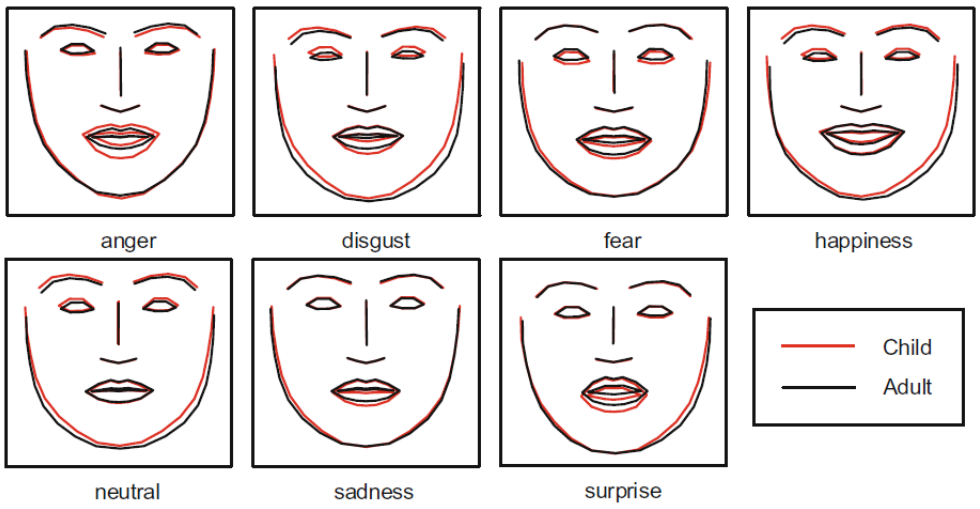

Figure 2.2: Average faces of children and adults when making different facial expressions [36]

	Anger	Disgust	Fear	Happiness	Neutral	Sadness	Surprise
Adults	84.47	75.37	75.51	95.19	76.14	69.34	92.71
Children	79.03	82.28	79.31	88.16	68.42	65.22	79.41

Figure 2.3: Classification results (%) on the balanced data. [36]

For the emotion classification, a binary classifier was trained. Positive and negative emotions were selected and the classifier was trained for each emotion individually. The algorithm classifies happiness with a much higher accuracy percentage of 88.16% while the lowest accuracy is for the sadness emotion which is around 65.22%. When tested on the adults dataset, the classifier performs much better on it than the children validation dataset as shown in Figure 2.3. The authors of the paper tried to use different datasets for expression detection of adults and children. But, for adult expression detection there needs to be more comprehensive and systematic approach to tackle the problem of achieving higher accuracy. With this reference, I tried to learn the importance of using different datasets and also do a comparative performance analysis among them. Also, it has been proven in [28] that a CNN based network can provide better accuracy than SVM.

The work by Ghaffar [9] provides an architecture for a CNN network with a face detection algorithm for better emotion detection accuracy. The paper worked on detection of the seven emotions which are neutral, anger, disgust, fear, happiness, sadness and surprise. The dataset was duplicated to obtain a much larger amount of training data. The architecture of the convolutional neural network used for the algorithm consisted of 3 layers, each of which is followed by a pooling layer and then three dense layers. The dropout rate of the dense layer is 20%. The data were segregated as 90% for training while 10% was used for testing. Datasets used were JAFFED and KDEF, and the accuracy achieved by combining the datasets is of 78%. Preprocessed images of size 80*100 is passed as input to the first layer of CNN.

Figure 2.4: Visualizing the 68 facial landmark coordinates [9]

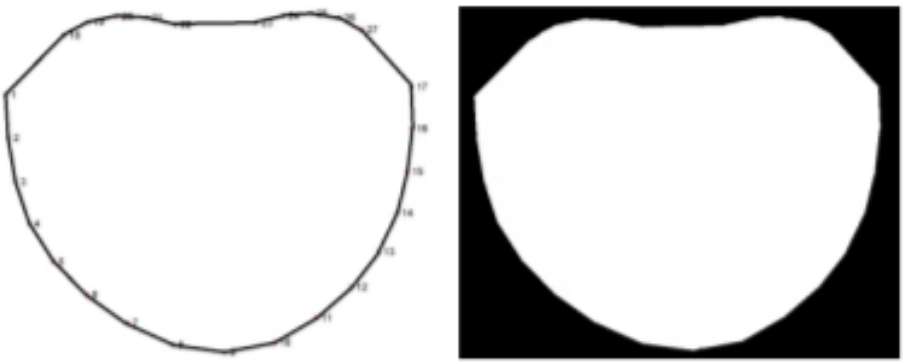

Figure 2.5: Convex hull of 0-27 facial landmarks and Mask obtained after filling of convex hull [9]

Features are extracted with higher accuracy from the images using intensity normalization and contrast enhancement. At first, face is detected in an image, area is estimated and the

face is extracted. Faces are detected using pre-trained model of facial landmark detector.The model finds the locations of 68 (x, y)-coordinates and each coordinate represents a region of interest on the face as shown in Figure 2.4. Using these coordinates, area of the face can be estimated. The next step is face extraction, which is done by extracting 1-27 points and applying convex hull to those points to create an enclosed structure as shown in Figure 2.5

Figure 2.6: Face Extraction [9]

The extracted mask is used to extract face area as shown in Figure 2.6

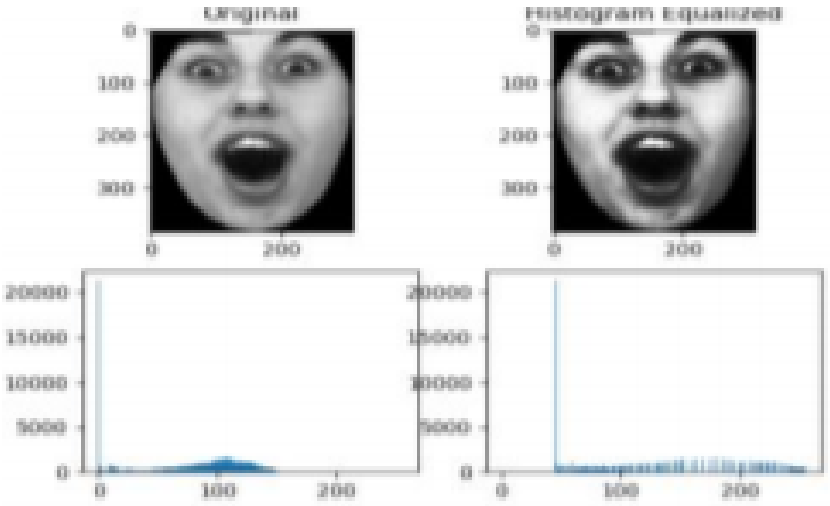

Figure 2.7: Histogram Equalization of Image [9]

The next step is that of image enhancement and duplication. Image enhancement is done by taking five copies of each image and applying histogram equalization as shown in Figure 2.7 or bilateral filter to the cropped face image. Bilateral image removes noise efficiently while preserving the edges as shown in Figure 2.8.

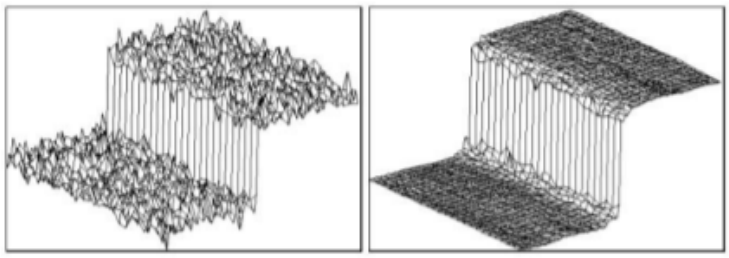

Figure 2.8: Bilateral Filter Noise Removal and Edge Preserving [9]

-1	-1	-1
-1	9	-1
-1	-1	-1

Figure 2.9: Kernel for Convolutional 2D Filter [9]

Other images of the copies are obtained by applying a convolutional 2D filter with kernel as shown in Figure 2.9 or by applying histogram equalization to the bilateral filtered image.

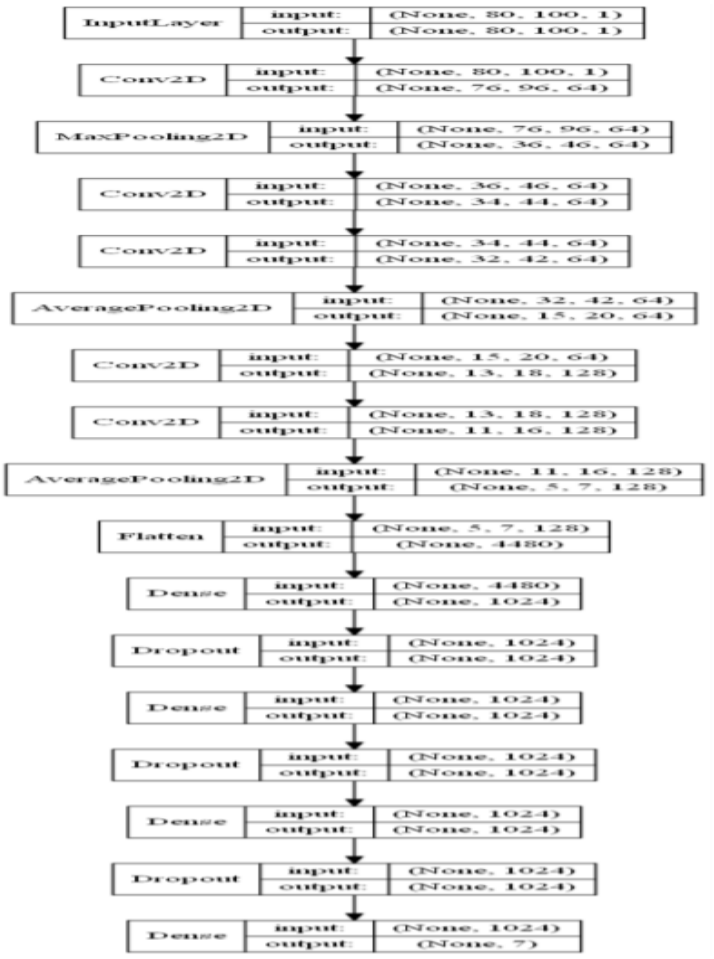

Figure 2.10: CNN Architecture [9]

All the images are resized to 80*100 dimensions and made to pass it into the Convolutional Neural Network with five layers as shown in Figure 2.10. This work gave the importance of a particular type of CNN network architecture with its learned parameters.

The authors in [27] built a SVM and CNN models capable of recognizing the seven basic emotions. The SVM model was that of one vs. all (OVA) [24] using sklearn's linear kernel SVM. Different methods were tried for the SVM model so as to increase the accuracy. First method was that of scaling the pixels so that each image had a mean pixel value of 0 and a variance of 1 and used the scaled pixel values as the new features. Next method which they

tried was that of using PCA (Principal Component Analysis) to isolate the most important components, this thereby reduced the dimensionality of the training set. The final method was that of (HOG) Histogram of Oriented Gradients to describe the distribution of gradients as different emotions have different gradients around the mouth and eye area. A combination of all these methods along with the SVM kernel increased the accuracy to 43.95%, which is comparatively low. The CNN model implemented in the paper is shown in Figure 2.11.

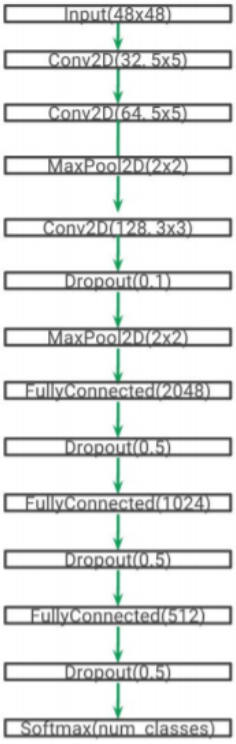

Figure 2.11: CNN Architecture [27]

It had an increased number of layers and decreased filter sizes to increase the number of parameters for a correct prediction on images which had smaller facial features. Overfitting in the model was handled by using dropout layers, early stopping around 100 epochs, and augmenting the training set. It obtained an accuracy of around 65%. The datasets used for training both the models were that of FER-2013 and CK+.

The above works acts as a major inspiration for the project baseline. The following is a three paragraph overview describing the related work of CNN using Tensorflow, Transfer learning using CNN and the Capsule Network in its subsequent paragraph.

Khorrami in [17] addressed the CK+ and the Toronto Face Dataset (TFD) by using zero-bias CNN to achieve state of the art emotion detection results. Mollahosseini [20] trained a single CNN for FER dataset using two convolution layers, one max pooling layer, and four "inception" layers i.e. sub-networks. The work in [19] created an attentional convolutional network, to classify the underlying emotion in the face images, thereby attending to the important face regions for emotion detection. The algorithm achieved an accuracy of 70%. Pramerdorfer in [26], created an ensemble based deep CNN network and achieved a state of the art test accuracy of 75.2% on FER2013 dataset. The ensemble method used here consisted of determining the class with the highest weighted mean score using the posterior class probabilities yielded from each individual with different weights.

The work in [16] used various pre-trained DCNN (deep CNN) models such as Alexnet, Resnet50, GoogLeNet, VGG-16, Resnet101, VGG-19, Inceptionv3, and InceptionResNetV2. The last few layers of each of the models were replaced to accommodate the brain MR (Magnetic Resonance) images. The test accuracy on the pre-trained AlexNet yielded the best performance of more than 90%.

The article [12] talks about implementing capsule network for classifying CK+ dataset. The images are processed using data augmentation before sending it through the network. Later, the classification results are combined with the Nao robot, who can visualize the emotion by changing its eyes colors. Most of these algorithms did not deal with smaller dataset and datasets having higher resolution images such as Ohio dataset. Also, my project tries to achieve higher accuracy and better performance than these algorithms by incorporating some of the unique properties of each research work.

Chapter 3

Methods

The project consists of comparing 3 main algorithms which will be analysed and tested with various datasets. Each of the algorithms has convolution as one of the layers as it has been proven that CNN (convolution neural network) models give better accuracy than SVM (Support Vector Machine) or Random Forrest [28]. These are the widely used, state of the art algorithms and each algorithm has a specialty.

1. Convolutional Neural Network in Tensorflow

   The basic architecture of a CNN (Convolutional Neural Network) is shown in Figure 3.1. A CNN consists of two types of layers:

   (a) Feature Extraction: Convolutions and Pooling

   (b) Classifier

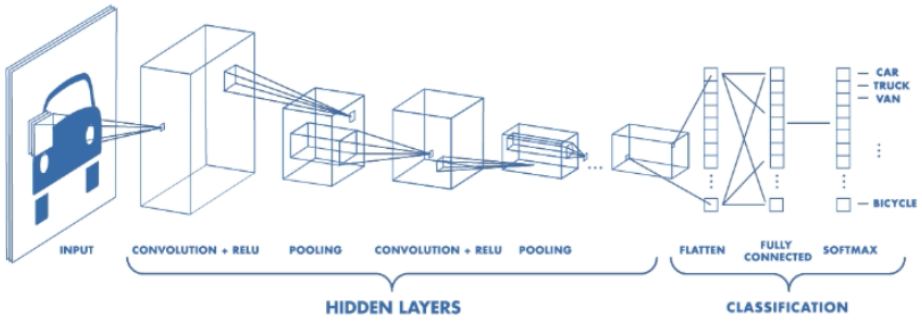

Figure 3.1: Basic Architecture of a CNN [32]

CNN uses a filter to get a feature map over the input images. Pooling technique is used for dimensionality reduction to reduce the number of parameters. There are numerous types of pooling techniques but the popular ones are max, average and median pooling. Average pooling takes the average in each window which decreases the feature map size while keeping the average of all the information. Dropout is a technique where randomly selected neurons are dropped during training to reduce overfitting. I have selected the Relu activation function over sigmoid as relu is more computationally efficient as it has better convergence performance than sigmoid [11]. Categorical cross-entropy is chosen as the loss function as cross-entropy calculates a score that summarizes the average difference between the actual and predicted probability distributions and categorical values ensures output as multi-class. The following Figure 3.2 is the CNN architecture which has been made. The last dense layer outputs a seven dimensional vector for the seven emotions.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 48, 48, 6)         456
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 24, 24, 6)         0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 16)        2416
_________________________________________________________________
activation_1 (Activation)    (None, 24, 24, 16)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 12, 12, 16)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 10, 10, 64)        9280
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 1600)              0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               204928
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 903
=================================================================
Total params: 217,983
Trainable params: 217,983
Non-trainable params: 0
```

Figure 3.2: Implemented CNN Architecture

2. FaceNet using Transfer Learning

FaceNet is a unified embedding for Face Recognition and Clustering. FaceNet learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. FaceNet embeddings can be used as feature vectors for face recognition, verification and clustering. For this project, the method can be used to extract the faces using the FaceNet_pytorch python library and applying transfer learning for expression detection. Transfer learning is a term which focuses on storing knowledge gained while solving one problem and applying it to a different but related problem [34]. Once the face is extracted, it is transformed to single channel

grayscale for emotion classifier as shown in Figure 3.3. The goal is to fine-tune a pre-trained ImageNet model for the datasets of CK+ and Ohio. The pre-trained model gives the training a good starting point.

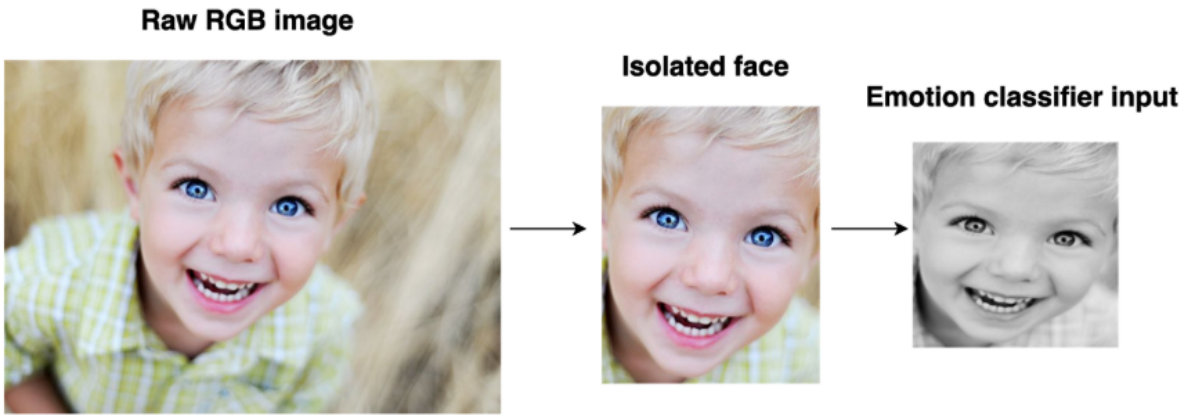

Figure 3.3: The first step crops the face from the raw image using an open-source, pre-trained face detection model. The second step resizes the image and transforms it to grayscale. [3].

RGB images (Ohio or CK+) are fed as input to the pretrained ResNet-50 model. ResNet (Residual Networks) tries to resolve the vanishing gradient problem in deep neural networks. The problem is tackled by using skip connections as shown in Figure 3.4. The skip connections help in mitigating the problem of vanishing gradient by allowing the alternate shortcut path for gradient to flow through and ensuring to learn an identity function making the higher layer work as good as the lower layer. ResNet-50 model is shown in Figure 3.5.

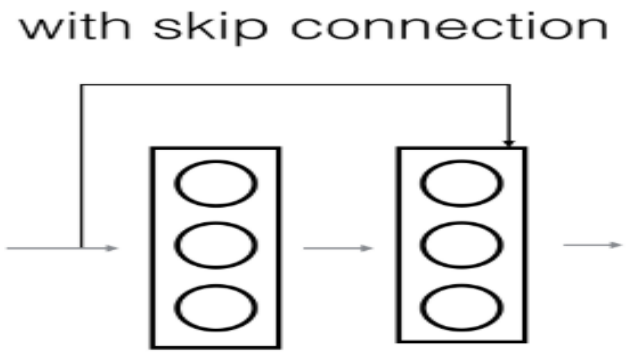

Figure 3.4: Skip Connection Image [5]

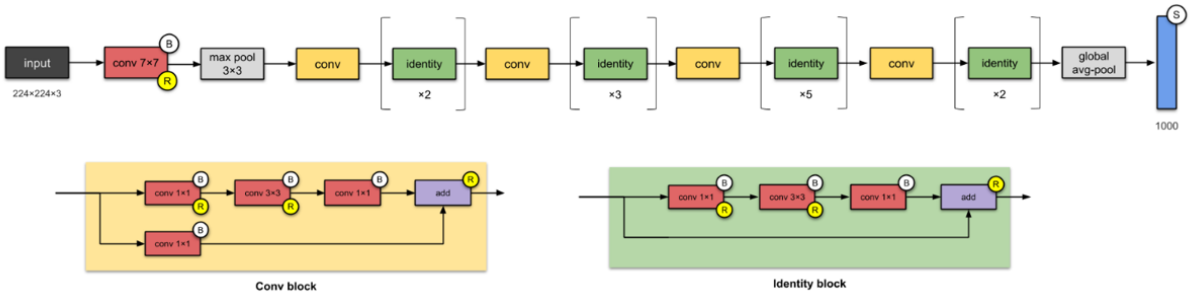

Figure 3.5: ResNet-50 architecture [15]

A new pretrained model is created by swapping the first convolution layer of ResNet-50 with a new initialized 1 channel convolution layer. This is the Gray ResNet-50 model. The difference between the output embeddings of the two ResNet models is computed as the L2 loss used in backpropagation to update the Gray ResNet-50 model as shown in Figure 3.6. The model is further finetuned using FER dataset and the output is seven-dimensional vector classifying emotions. Gray images from CK+ or Ohio dataset again finetunes the model with last eleven layers as frozen. The layers are kept as frozen as the number of images are limited and the number (eleven) gave the highest accuracy. The model thereby learns from richer features present in the "wild" (our) dataset. Grayscale images are used here because the model should focus on the actual facial expression and not learn any biases that may come with color.

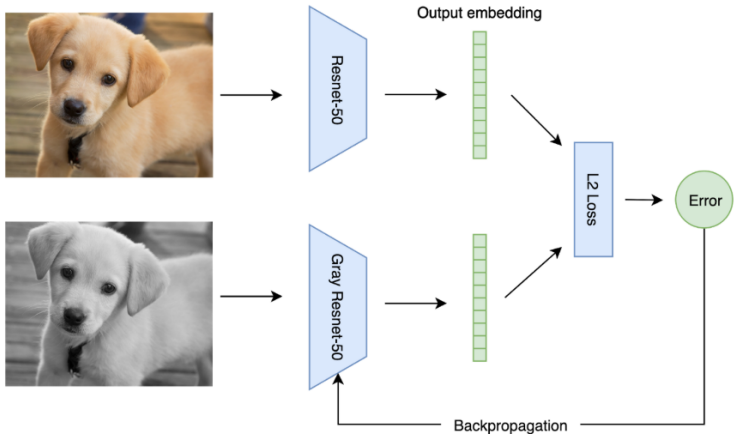

Figure 3.6: Fine-tuning Resnet-50 using grayscale versions of ImageNet [3].

3. Capsule Network

(a) Problem with Convolutional Neural Network

The use of pooling layer in CNN loses a lot of valuable information and they ignore relation between part and the whole. Take for instance, a CNN used as a face detector in images. However, a CNN would have learnt that there are five fundamental features that make up a face (two eyes, nose, mouth, and the oval shape of a face). However, there could be an image that has an oval shaped figure, with two eyes, a nose and a mouth all lying outside this oval shaped figure as shown in Figure 3.7. Then, the CNN will classify it also as a face. This issue of max-pooling causes incorrect results related to application for image classification. The capsule network maintains the hierarchical pose between the parts/features of the object in an image. Capsule network has a base on the CNN, but the neuron form is converted from scalar to the vector which contains highly informative outputs. [30].

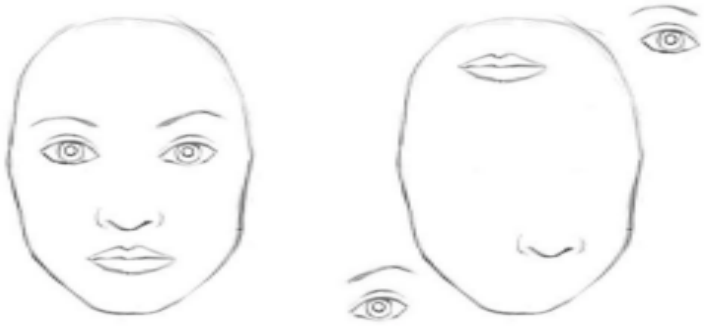

Figure 3.7: To a CNN, both pictures are similar, since they both contain similar elements. [22]

(b) Difference Between Neurons and Capsules

The length of the output/activity vector represents the probability that the entity exists, and its orientation represents the instantiation parameters. The state of that particular detected feature is encoded as the direction in which that vector points to. Therefore, when a detected feature moves around the image or its state somehow changes, the probability still stays the same (i.e its length) but its orientation changes as shown in Figure 3.8 when the eye orientation changed.

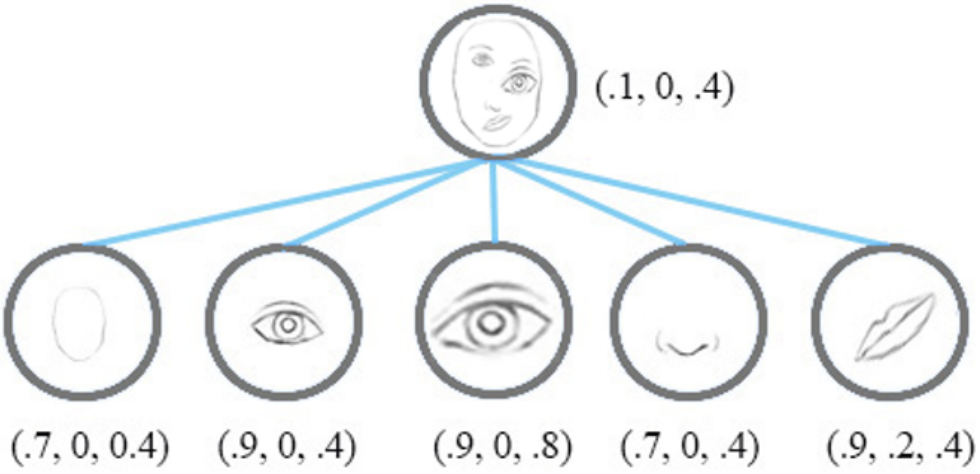

Figure 3.8: Change of activity vector as orientation changes. [22]

In CNNs, a neuron receives the scalar inputs from other lower layer neurons. The inputs are multiplied by scalar weights which then get summed up. The sum is passed to the activation function which outputs another scalar value according to its functionality. The weights are later fine-tuned using the backpropagation algorithm, to match the target output values. The calculations happening the capsule network vs. the neurons is shown in Figure 3.9

Capsule vs. Traditional Neuron			
Input from low-level capsule/neuron		vector($\mathbf{u}_i$)	scalar(x_i)
Operation	Affine Transform	$\hat{\mathbf{u}}_{j\|i} = \mathbf{W}_{ij}\mathbf{u}_i$	–
	Weighting	$\mathbf{s}_j = \sum_i c_{ij}\hat{\mathbf{u}}_{j\|i}$	$a_j = \sum_i w_i x_i + b$
	Sum		
	Nonlinear Activation	$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1+\|\mathbf{s}_j\|^2}\frac{\mathbf{s}_j}{\|\mathbf{s}_j\|}$	$h_j = f(a_j)$
Output		vector($\mathbf{v}_j$)	scalar(h_j)

Figure 3.9: The Difference between Neurons and Capsules. [23]

Capsule network works on the following four steps as mentioned below:

i. Matrix Multiplication of Input Vectors

   The probability vectors of features detected are multiplied by their corresponding weight matrices (these weight matrices encode important spatial relationships between lower level features and higher level features). After the multiplication operation is done, the predicted position of the higher level feature is obtained.

ii. Scalar Weighting of Input Vectors

These weights are learned using the Dynamic Routing Algorithm as described in section 3c. The multiplied input vectors are multiplied by the weights.

iii. Sum of weighted Input Vectors

All of the weighted input vectors are added to produce a single sum.

iv. Vector to Vector Non-linearity.

Vector-to-Vector non-linearity, also known as Squash, takes a vector and squashes it to have a length of no more than 1 without changing its direction. All these steps are shown in Figure 3.10

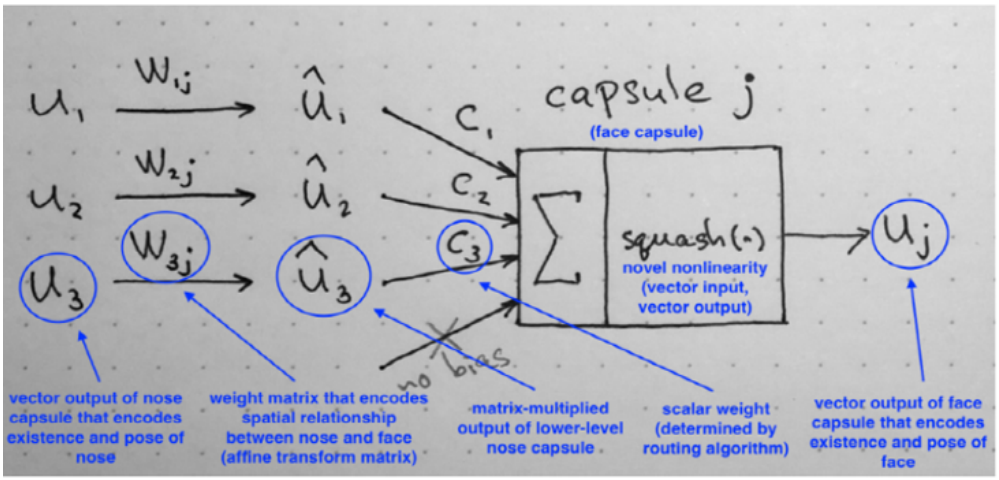

Figure 3.10: Computations inside of a capsule. [23]

(c) Dynamic Routing Dynamic routing decides where each capsule output goes. As shown in Figure 3.11, the lower capsules decide which higher capsule it wants to send its output to. It will make its decision by adjusting the weights (C), which are multiplied with the lower capsule's output before sending it to either left or right higher-level capsules J and K. The higher level capsules receive many input vectors from other lower-level capsules. All these inputs are represented by red and blue points. Where these points cluster together, this means that predictions of lower level capsules are close to each other. Thus, after multiplying with matrix

W, it lands away from prediction in capsule J and close to prediction in capsule K. Thus capsule K accommodates the target result well, thereby adjusting its weights.

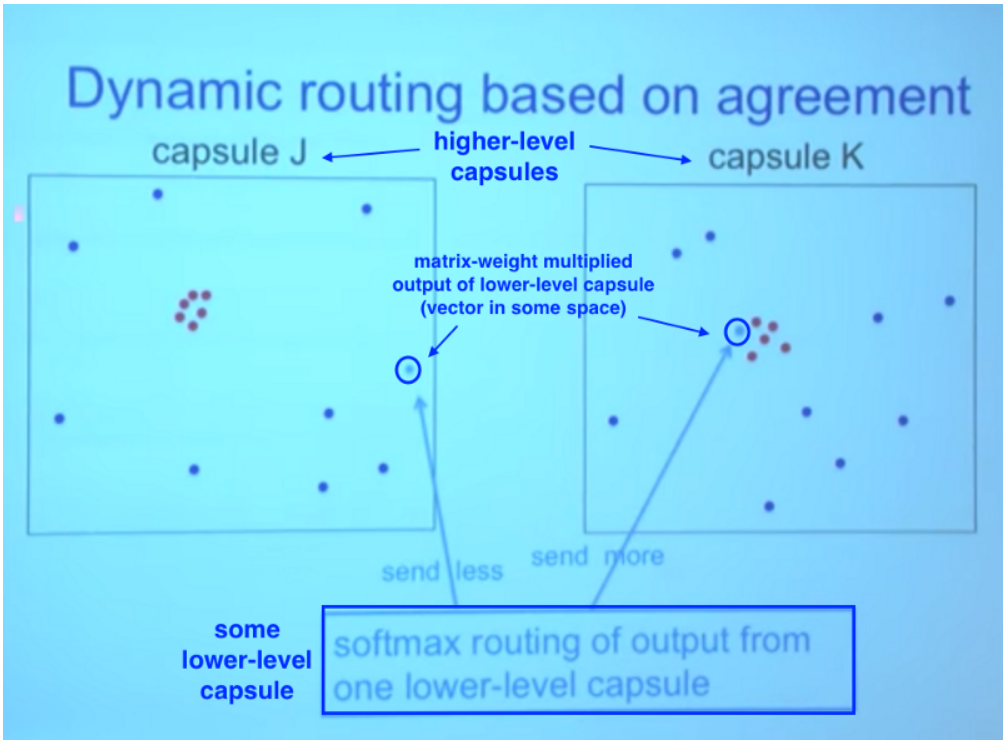

Figure 3.11: Lower level capsule will send its input to the higher level capsule that "agrees" with its input. [23]

(d) Capsule Architecture

The Architecture has 2 components: Encoder and Decoder.

i. Encoder

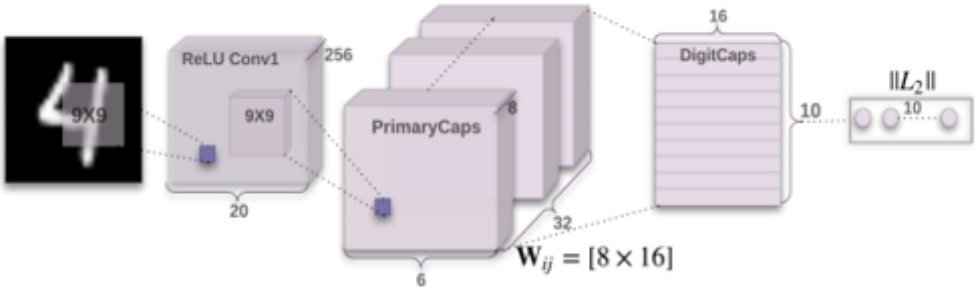

Figure 3.12: CapsNet encoder architecture. [30]

As shown in Figure 3.12, takes an image as input and learns to encode it into vector of instantiation parameters. It consists of three layers which are the Convolutional Layer, PrimaryCaps Layer and the DigitCaps Layer. The PrimaryCaps Layer take basic features detected by the convolutional layer and produce combinations of the feature. The DigitCaps Layer outputs a 16x10 matrix. This layer has seven digit capsules (one for each emotion). For each training example, one loss value will be calculated for each of the seven vectors using the loss function as shown in Figure 3.13.

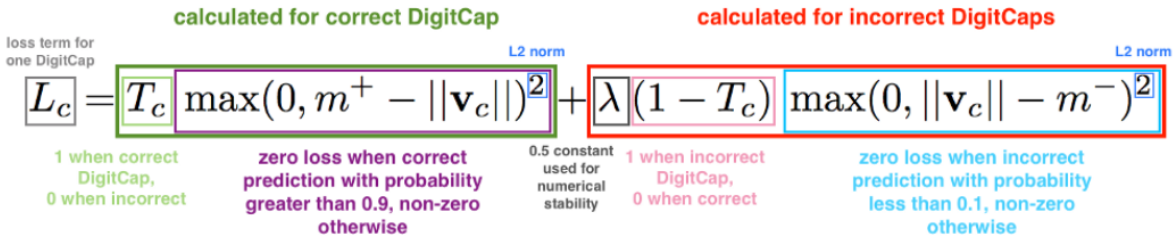

Figure 3.13: CapsNet Loss Function. [30]

ii. Decoder

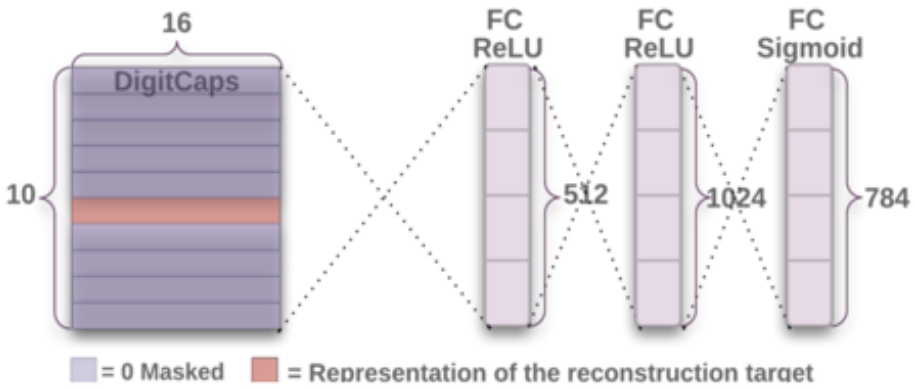

Figure 3.14: CapsNet decoder architecture. [30]

As shown in Figure 3.14, the decoder takes a vector from the DigitCaps layer and learns to decode it into an image i.e. reconstruct the image. Decoder is used as a regularizer that takes output of the correct DigitCap as input and learns to recreate a 48 by 48 pixel image, with the loss function being Euclidean ditance between the reconstructed image and the input image. It has three fully-connected layers which produces an output vector of length 2304 which is then reshaped to give back a 48x48 decoded image.

Chapter 4

Experiments

Each dataset has been divided into training and validation set. The purpose of training set is to build and train the model by adjusting its weights to make a more accurate prediction in the future and with each iteration the process refines what the network has 'learned'. The validation set is to test the neural network after it has been trained, here in this case it is a subset of the original dataset. The validation is a separate set of images which was not used for training, thereby representing the new scenarios. The training accuracy describes how well the neural network is predicting emotions from samples in the training set. The validation accuracy shows how well the neural network is predicting emotions from samples in the validation set. Training accuracy is usually higher than validation accuracy. This is because the neural network learns patterns from the training samples which may not be present in the validation set. The training and the validation set is split in the ratio of 85:15. The results and the graphs obtained by applying the three algorithms for the (CK+) Extended Cohn-Kanade and the Ohio dataset is described below. FER2013 is the base dataset for the methods implemented in section 3. Also, the explanation of a particular result obtaining is also described in detail. For the classification report, precision, recall and F1-score is calculated. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Recall is the ratio of correctly predicted positive observations to the all observations in actual class. F1 Score is the weighted average of Precision and Recall. The confusion matrix is the error matrix in a specific table layout,

allowing the visualization of the results. The labels for the confusion matrix is as follows:

- 0 : Angry
- 1 : Disgust
- 2 : Fear
- 3 : Happy
- 4 : Sad
- 5 : Surprise
- 6 : Neutral

4.1 Background Experiment

I tried to run the repository of 'Facial expression recognition using CNN' [10] with model already trained on FER dataset. I also ran the code to predict emotions from real-time webcam. The following result is obtained as shown in Figure 4.1. The predicted emotion is always happy or neutral on the real-time webcam feed images.

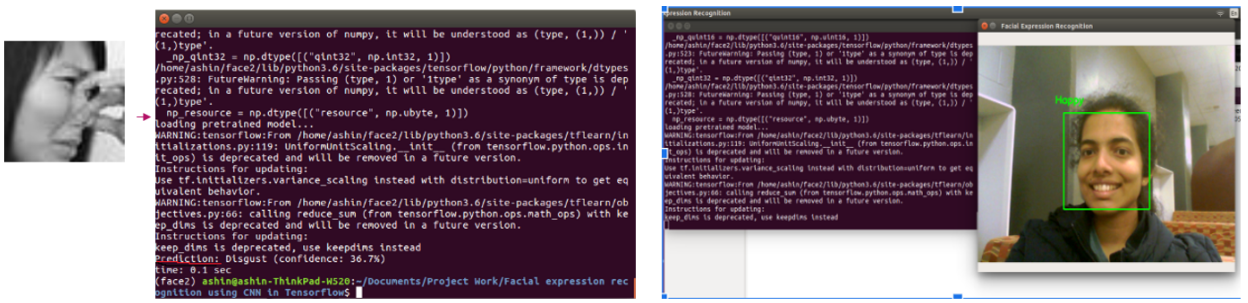

Figure 4.1: Output predicted emotions on running [10] code repo

4.2 Convolutional Neural Network in Tensorflow

1. CK+ Dataset

   The training accuracy obtained is 82.35% and the validation accuracy is 76.35%.

   The following Figure 4.2 shows the predictions on the images.

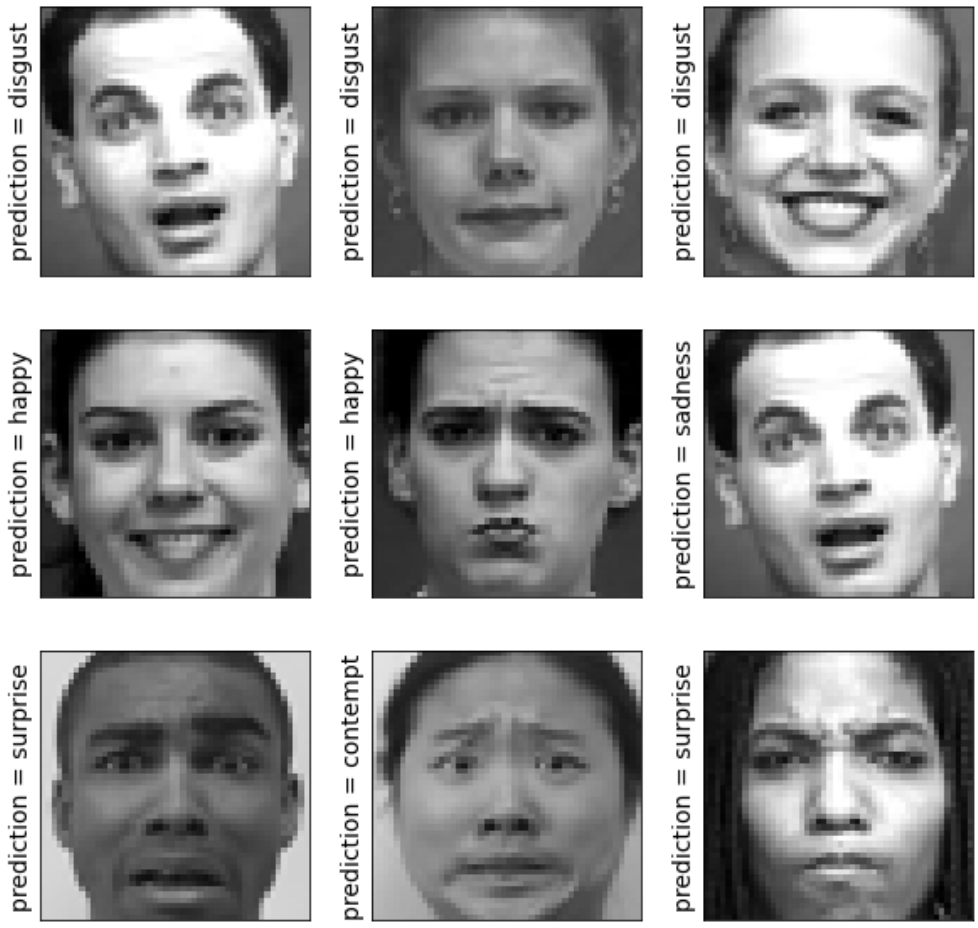

Figure 4.2: Output predicted emotions on Cohn-Kanade Dataset

The validation accuracy obtained is pretty good enough as per the paper [26], the validation accuracy obtained by the CNN model of the authors trained and tested on FER dataset is 75.2%.

The graphs of Training loss vs Epochs and Validation accuracy vs Epochs is shown in Figure 4.3:

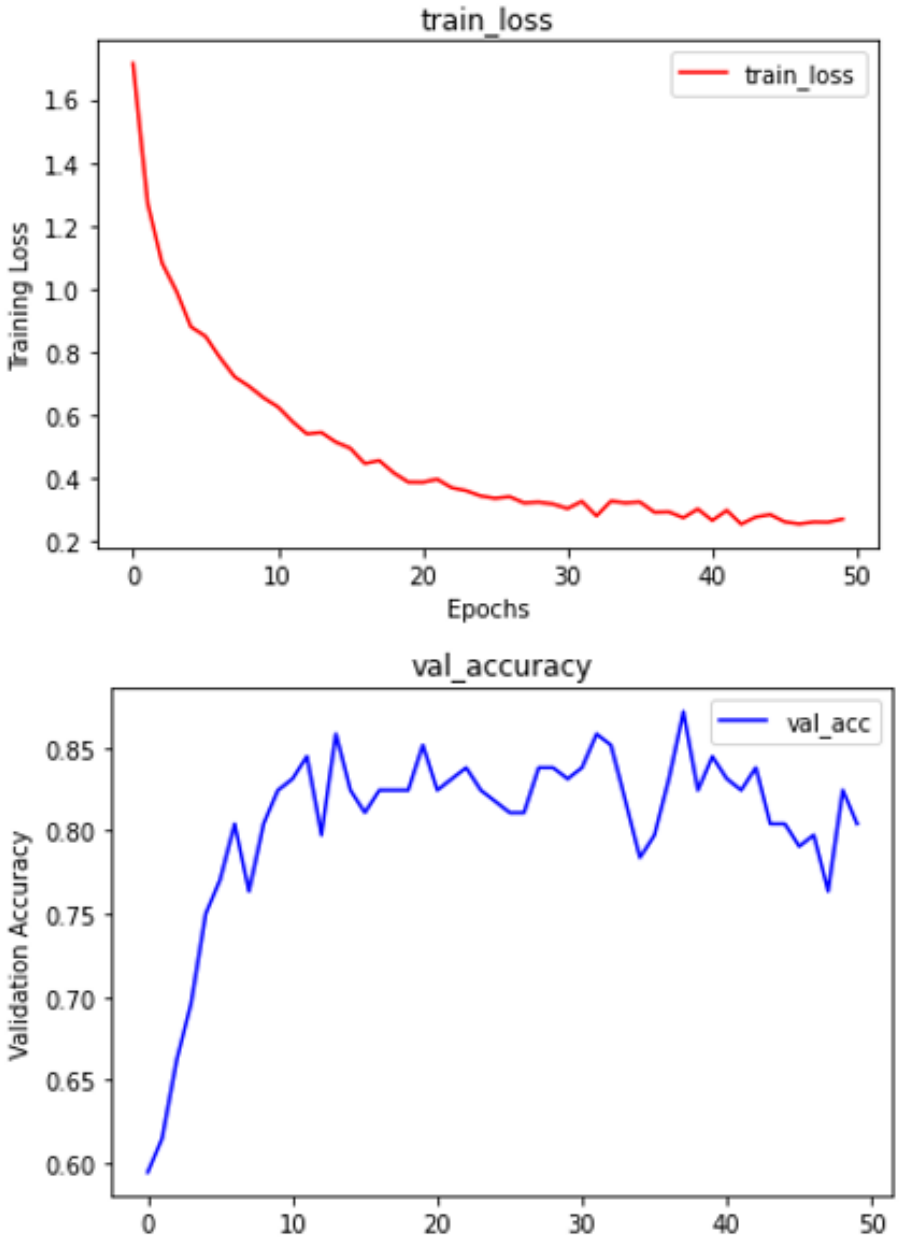

Figure 4.3: Training loss and Validation accuracy graphs

The validation accuracy fluctuates a lot but it remains on the higher side of greater than 75%.

2. Ohio Dataset

The training accuracy obtained is 92.4% and the validation accuracy is 33.33%. The following Figure 4.4 shows the predictions on the images.

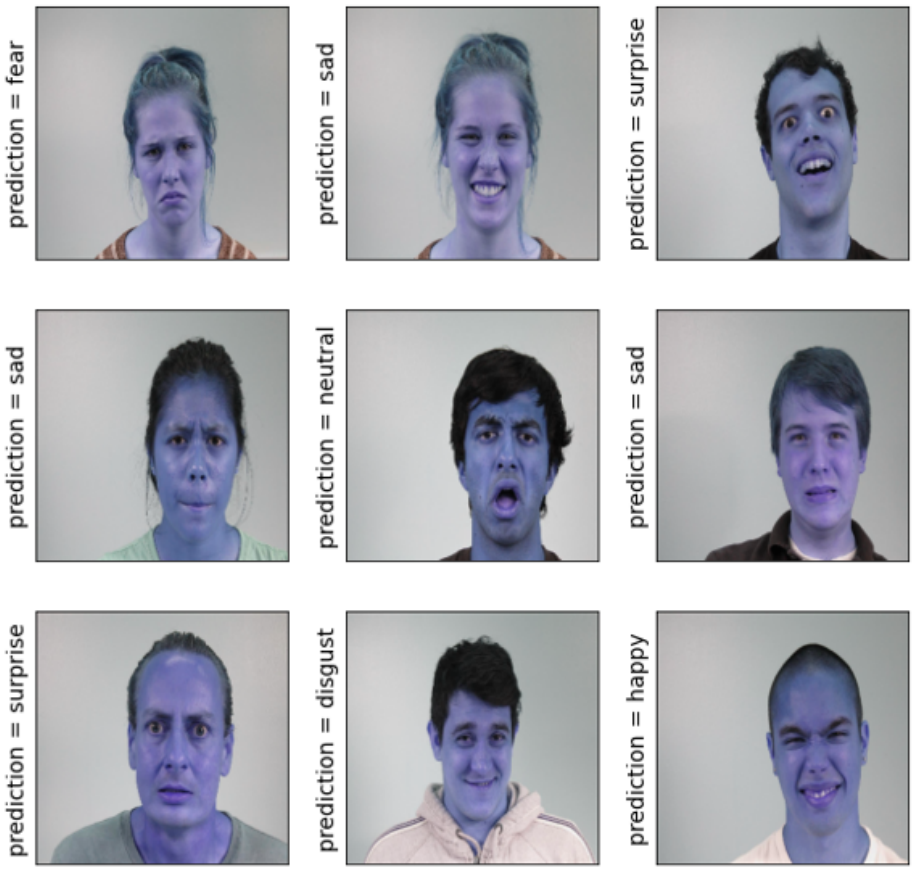

Figure 4.4: Output predicted emotions on Ohio Dataset

The graphs of Training loss vs Epochs and Validation accuracy vs Epochs are shown in Figure 4.5:

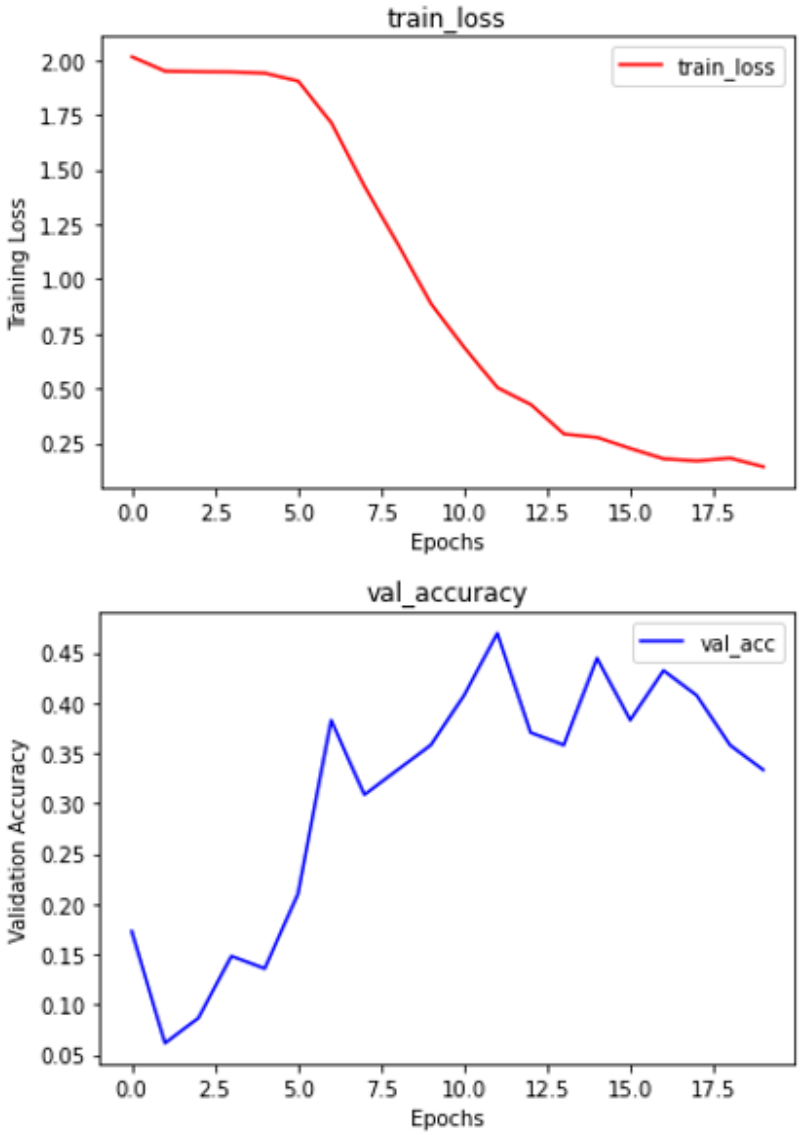

Figure 4.5: Training loss and Validation accuracy graphs

The low validation accuracy versus the high training accuracy accuracy in this case can be related to the reason of overfitting. Machine learning models suffer from a problem of overfitting, where the model learns the parameters so well that it is unable to adjust to the new scenarios resulting in high variance. The overfitting can be dealt by fine-tuning the architecture of the CNN; some examples include finding and removing

redundant parameters, adding new parameters in more useful places in the structure, adjusting the learning rate decay schedule, adapting the location and probability of dropout and experimenting to find ideal stride sizes.

4.3 FaceNet using Transfer Learning

1. Ohio Dataset

The validation accuracy is 42.91%. The graphs of Training loss vs Epochs and Validation accuracy vs Epochs are shown in Figure 4.6 and Figure 4.7:

Figure 4.6: Training loss vs Epochs

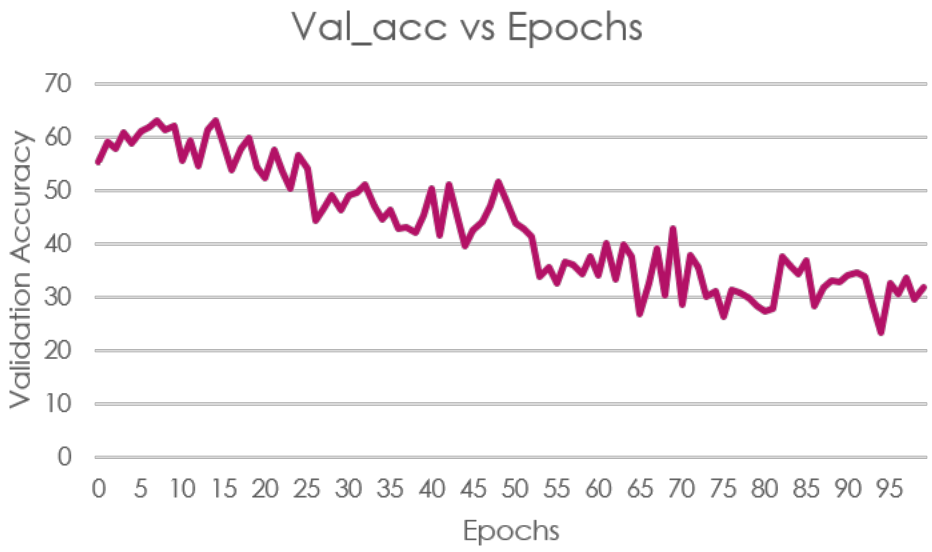

Figure 4.7: Validation Accuracy vs Epochs

If the graph 4.7, is observed it can be seen that the validation accuracy reached a good high accuracy at an epoch of around 50. This is called the optimum model complexity as shown in Figure 4.8. After this point, the model starts overfitting.

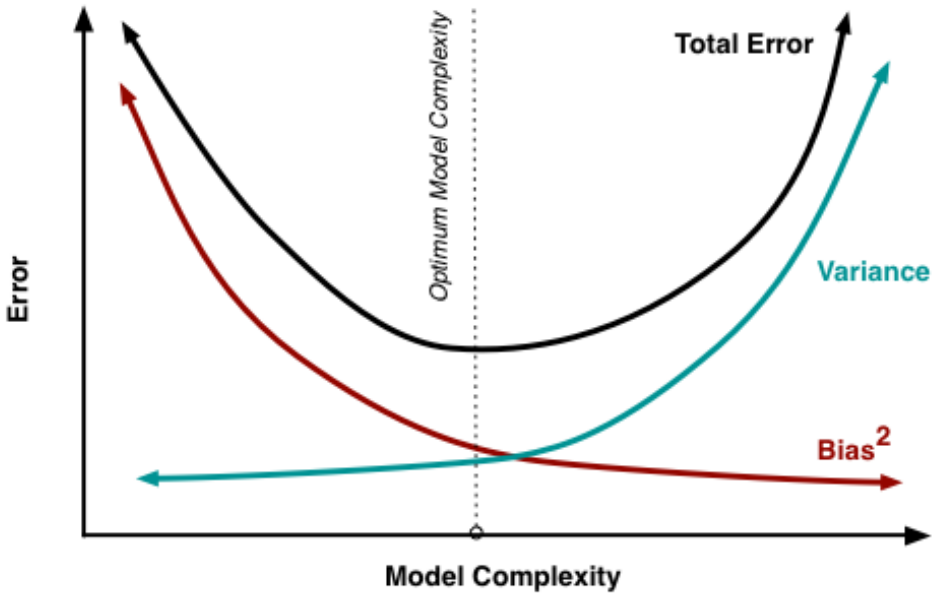

Figure 4.8: Bias and variance contributing to total error [8]

4.4 Capsule Network

Capsule network causes underfitting when lesser dataset images are used for training [12]. To overcome this issue, data augmentation is done by rotating, amplifying and adding with the salt and pepper noise. After augmenting the CK+ dataset had a total number of images to 157885 and the Ohio dataset to 21134.

1. CK+ Dataset

   The training accuracy obtained is 99.7% and the validation accuracy is 99.3%. The classification report using the ground truth and the predicted labels of neutral + six basic emotions is shown in Figure 4.9

	precision	recall	f1-score
Angry	1.00	0.89	0.94
Disgust	1.00	1.00	1.00
Fear	1.00	1.00	1.00
Happy	1.00	1.00	1.00
Sad	1.00	1.00	1.00
Surprise	0.93	1.00	0.97
Neutral	1.00	1.00	1.00
accuracy			0.99
macro avg	0.99	0.98	0.99
weighted avg	0.99	0.99	0.99

Figure 4.9: Classification Report for the CK+ Dataset

The confusion matrix is shown in Figure 4.10.

```
[[1621    4    0    0    0   11    2]
 [   0 1355    0    0    1    4    0]
 [   3    0 4548    0    0    0    0]
 [   0    3    0 1831    0    1    2]
 [   0    0    0    0 5322    0    0]
 [   4    0    0    1    0 2093    0]
 [   0    0    0    0    0    0 6194]]
```

Figure 4.10: Confusion Matrix for the CK+ Dataset

The graphs of Training loss vs Epochs and Validation accuracy vs Epochs are shown in Figure 4.11:

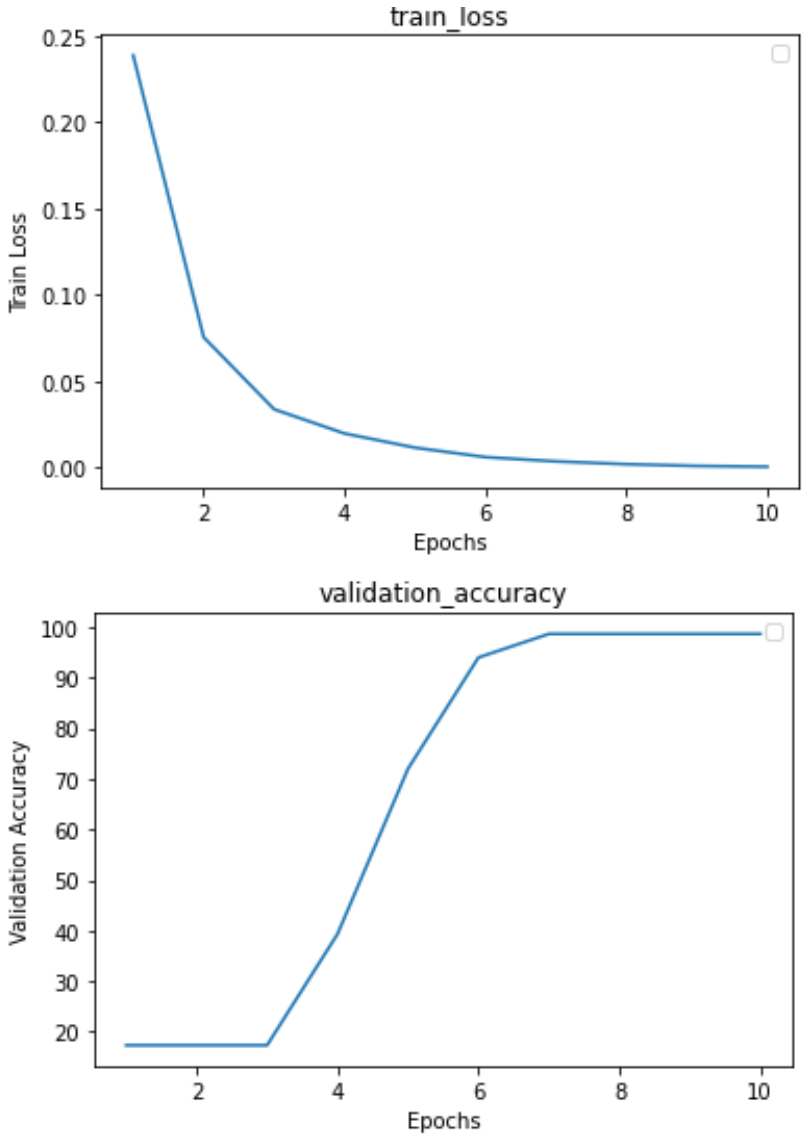

Figure 4.11: Training loss and Validation accuracy graphs

2. Ohio Dataset

The training accuracy obtained is 97.3% and the validation accuracy is 96.2%. The classification report using the ground truth and the predicted labels of neutral + six basic emotions is shown in Figure 4.12.

	precision	recall	f1-score	support
Angry	0.95	0.97	0.96	1174
Disgust	1.00	0.85	0.92	693
Fear	0.99	0.96	0.97	823
Happy	0.98	1.00	0.99	830
Sad	0.00	0.00	0.00	0
Surprise	0.90	0.98	0.94	858
Neutral	0.98	1.00	0.99	822
micro avg	0.96	0.96	0.96	5200
macro avg	0.83	0.82	0.82	5200
weighted avg	0.96	0.96	0.96	5200

Figure 4.12: Classification Report for the Ohio Dataset

The confusion matrix is shown in Figure 4.13.

```
[[1023    7    0    0    0   41    3]
 [  24  591   11    1    0   70    0]
 [  23    0  806    4    0   10   12]
 [   2    1    3  851    0    2    0]
 [   0    0    0    0    0    0    0]
 [   6    0    2    0    0  867    0]
 [   0    0    2    0    0    0  838]]
```

Figure 4.13: Confusion Matrix for the Ohio Dataset

The graphs of Training loss vs Epochs and Validation accuracy vs Epochs are shown in Figure 4.14:

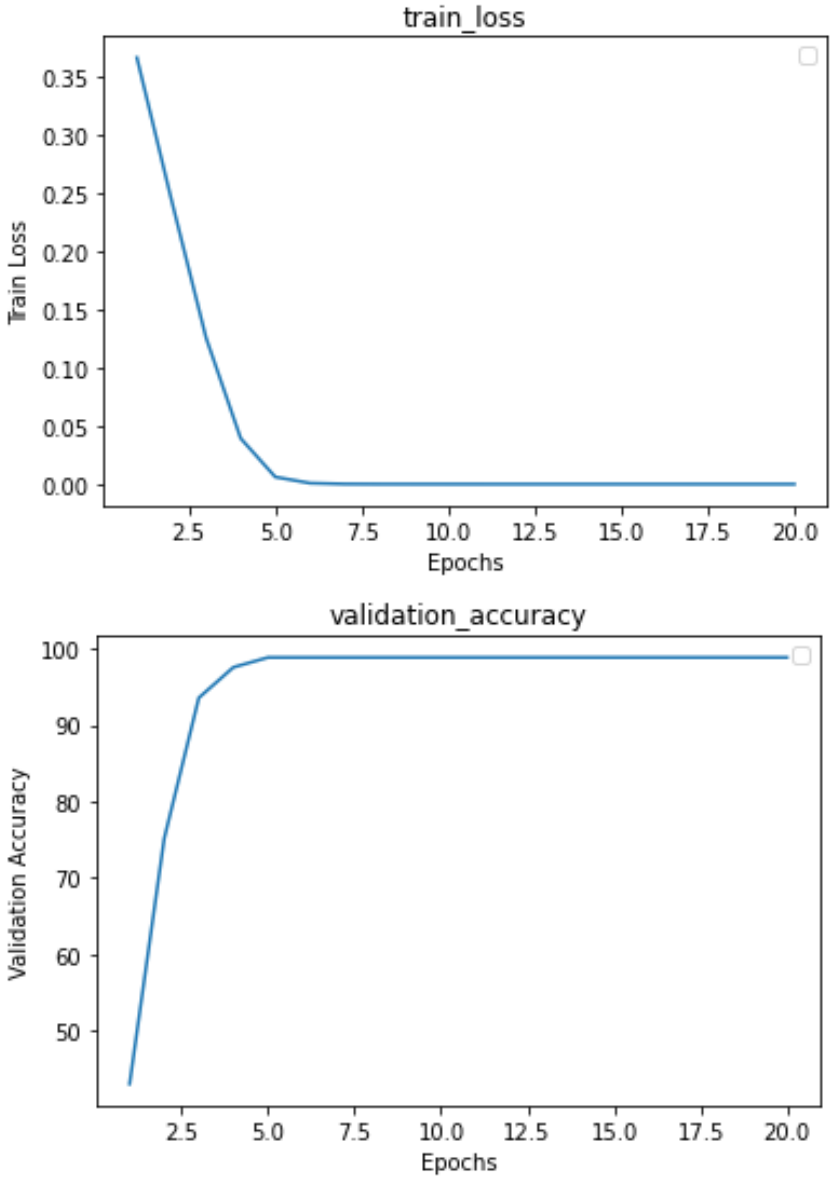

Figure 4.14: Training loss and Validation accuracy graphs

Chapter 5

Summary and Discussion

5.1 Summary

The results can be summarized as follows:

Table 5.1: Summary Results

Algorithm	Datasets	Train Accuracy	Validation Accuracy
Convolution Neural Network in Tensorflow	*FER*		*75.2%* [26]
	CK+	82.35%	76.35%
	Ohio	92.4%	33.33%
FaceNet using Transfer Learning	Ohio		42.9%
Capsule Network	*FER*		*94.37%*
	CK+	99.7%	99.3%
	Ohio	97.3%	96.2%

5.2 Discussion

Rigorous work is needed to implement and evaluate a facial emotion detection system which can be used real-time. These systems can play a vital role in semi-automated vehicles. In this

work, I studied different adult facial expression recognition methods as in section 2. Most of the algorithms were unable to tackle smaller datasets and datasets having high resolution images. They also did not achieve accuracy as high as 90%. In addition, I tried to come up with a comparative analysis to compare the implemented expression recognition methods using the validation accuracy on the Ohio and CK+ dataset as it can be seen in Table 5.1.

The important result is that the capsule network has higher accuracy percentages than other methods. Thereby, it can classify emotions much accurately. This is due to the advantage of capsule network taking vectors as input instead of scalar values as in a fully connected convolution layers. The issues due to max pooling are also resolved in the capsule network. Algorithms trained and validated on CK+ (extended cohn-kanade) dataset is shown to have higher accuracy due to it being grayscale images which captures features better. Also CK+ dataset has a small number of subjects(123) which are being posed and in a centered form with good lighting. On the other hand, Ohio and FER-2013 dataset captures more accurately reflect real-time conditions due to its automatically captured, non-posed photos.

5.2.1 Limitations

System limitations:

1. To run the FaceNet using Transfer learning, it took lots of days on the ARC cluster. FaceNet based model has considerable amount of computations which are happening making the process time consuming.

2. CUDA out of memory error occurs for a larger and higher resolution datasets on the ARC cluster.

3. Labelled dataset is not publicly available. I had to manually sort the CK+ and the

Ohio dataset in its labelled emotion folder.

4. Capsule Network gives better performance on a larger dataset, so data augmentation is required for smaller datasets.

Experimental limitations:

1. The algorithms are not tested with real human participants.

2. It is not clear about what exactly the background preprocessing that were done for the FER results in other research works.

5.2.2 Future Work

For the future work, I would like to do a similar comparison such as creating a confusion matrix or a classification report for all the three algorithms. More emotion recognition algorithms can be compared such as Emopy [25]. I will try to combine the emotion affect system in the Nervtech simulator and study the results of real-time emotion detection, similar to the work done for Nao robot [12]. I will also try to fix the overfitting of the CNN in Tensorflow model.

5.3 Conclusion

Through this Masters project, I learnt newer algorithms such as capsule networks and transfer learning using CNN. Also, I learnt new techniques to handle smaller datasets. This project tries to do a comparative study of different human emotion detection algorithms. Based on

the observation, the capsule network outperformed the CNN in Tensorflow and the Transfer learning with CNN.

Bibliography

[1] Challenges in representation learning: Facial expression recognition challenge. https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data/.

[2] behavioralsignals.com. What is emotion ai? https://behavioralsignals.com/what-is-emotion-ai/.

[3] Martin Chobanyan. Training an emotion detector with transfer learning. https://towardsdatascience.com/training-an-emotion-detector-with-transfer-learning-91dea84adeed.

[4] Shichuan Du, Yong Tao, and Aleix M Martinez. Compound facial expressions of emotion. *Proceedings of the National Academy of Sciences*, 111(15):E1454–E1462, 2014.

[5] Priya Dwivedi. Understanding and coding a resnet in keras. https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33.

[6] Paul Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992.

[7] Mike Elgan. What happens when cars get emotional? https://www.fastcompany.com/90368804/emotion-sensing-cars-promise-to-make-our-roads-much-safer.

[8] Scott Fortmann-Roe. Understanding the bias-variance tradeoff. http://scott.fortmann-roe.com/docs/BiasVariance.html.

[9] Faisal Ghaffar. Facial emotions recognition using convolutional neural net. *arXiv preprint arXiv:2001.01456*, 2020.

[10] Amine Horseman. Facial expression recognition using cnn in tensorflow. https://github.com/amineHorseman/facial-expression-recognition-using-cnn.

[11] RockTheStar (https://stats.stackexchange.com/users/41749/rockthestar). What are the advantages of relu over sigmoid function in deep neural networks? Cross Validated. URL https://stats.stackexchange.com/q/126238. URL:https://stats.stackexchange.com/q/126238 (version: 2019-10-24).

[12] Vipul Jain, Srikanta Patnaik, Florin Popențiu Vlădicescu, and Ishwar K Sethi. Recent trends in intelligent computing, communication and devices.

[13] Myounghoon Jeon. Don't cry while you're driving: sad driving is as bad as angry driving. *International Journal of Human–Computer Interaction*, 32(10):777–790, 2016.

[14] Myounghoon Jeon, Bruce N Walker, and Jung-Bin Yim. Effects of specific emotions on subjective judgment, driving performance, and perceived workload. *Transportation research part F: traffic psychology and behaviour*, 24:197–209, 2014.

[15] Raimi Karim. Illustrated: 10 cnn architectures. https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d.

[16] Taranjit Kaur and Tapan Kumar Gandhi. Deep convolutional neural networks with transfer learning for automated brain image classification. *Machine Vision and Applications*, 31:1–16, 2020.

[17] Pooya Khorrami, Thomas Paine, and Thomas Huang. Do deep neural networks learn facial action units when doing expression recognition? In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 19–27, 2015.

[18] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain 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.

[19] Shervin Minaee and Amirali Abdolrashidi. Deep-emotion: Facial expression recognition using attentional convolutional network. *arXiv preprint arXiv:1902.01019*, 2019.

[20] Ali Mollahosseini, David Chan, and Mohammad H Mahoor. Going deeper in facial expression recognition using deep neural networks. In *2016 IEEE Winter conference on applications of computer vision (WACV)*, pages 1–10. IEEE, 2016.

[21] Ram Krishna Pandey, Souvik Karmakar, AG Ramakrishnan, and Nabagata Saha. Improving facial emotion recognition systems using gradient and laplacian images. *arXiv preprint arXiv:1902.05411*, 2019.

[22] Max Pechyonkin. Understanding hinton's capsule networks. part 1. intuition., . https://pechyonkin.me/capsules-1/.

[23] Max Pechyonkin. Understanding hinton's capsule networks part 2. how capsules work., . https://pechyonkin.me/capsules-2/.

[24] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[25] Angelica Perez. Emopy: a machine learning toolkit for emotional expression. https://www.thoughtworks.com/insights/blog/emopy-machine-learning-toolkit-emotional-expression.

[26] Christopher Pramerdorfer and Martin Kampel. Facial expression recognition using convolutional neural networks: state of the art. *arXiv preprint arXiv:1612.02903*, 2016.

[27] M Quinn, Grant SIVESIND, and Guilherme REIS. Real-time emotion recognition from facial expressions, 2015.

[28] Sebastian Raschka. When does deep learning work better than svm or random forrests?, 2016. https://www.kdnuggets.com/2016/04/deep-learning-vs-svm-random-forest.html.

[29] Limuel Z Ruiz, Renmill Patrick V Alomia, A Dominic Q Dantis, Mark Joseph S San Diego, Charlymiah F Tindugan, and Kanny Krizzy D Serrano. Human emotion detection through facial expressions for commercial analysis. In *2017IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, pages 1–6. IEEE, 2017.

[30] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017.

[31] Harsh Sanghavi. Measuring the influence of anger on takeover performance in semi-automated vehicles. 2020. Unpublished Master's Thesis.

[32] Nishank Sharma. How to do facial emotion recognition using a cnn? https://medium.com/themlblog/how-to-do-facial-emotion-recognition-using-a-cnn-b7bbae79cd8f.

[33] Yong Tao Shichuan Du and Aleix M. Martinez. Compound facial expressions of emotion. https://www.pnas.org/content/111/15/E1454.

[34] Wikipedia contributors. Transfer learning — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Transfer_learning&oldid=949614116, 2020. [Online; accessed 14-April-2020].

[35] Chris Wiltz. Emotional ai makes your car really know how you feel. https://www.designnews.com/automation-motion-control/emotional-ai-makes-your-car-really-know-how-you-feel/86552748258601.

[36] Zhi Zheng, Xingliang Li, Jaclyn Barnes, Chung-Hyuk Park, and Myounghoon Jeon. Facial expression recognition for children: Can existing methods tuned for adults be adopted for children? In *International Conference on Human-Computer Interaction*, pages 201–211. Springer, 2019.