PROJECT REPORT ON

Investigating Deep Learning Techniques for Facial Expression Recognition

SUBMITTED BY

ASEEM APASTAMB (402031)
PUSHKAR GANORKAR (402057)
SARVESH KHIRE (402082)

UNDER THE GUIDANCE OF
Mrs. Suja Panickar

Department Of Computer Engineering
MAEER's MAHARASHTRA INSTITUTE OF TECHNOLOGY
Kothrud, Pune 411 038
2019-2020

MAHARASHTRA ACADEMY OF ENGINEERING AND EDUCATIONAL RESEARCHES

MAHARASHTRA INSTITUTE OF TECHNOLOGY

PUNE

DEPARTMENT OF COMPUTER ENGINEERING

C E R T I F I C A T E

This is to certify that

Aseem Apastamb (402031)

Pushkar Ganorkar (402057)

Sarvesh Khire (402082)

of B. E. Computer successfully completed project report in

INVESTIGATING DEEP LEARNING TECHNIQUES FOR FACIAL EXPRESSION RECOGNITION

to my satisfaction and submitted the same during the academic year 2019-2020 towards the partial fulfillment of degree of Bachelor of Engineering in Computer Engineering of Pune University under the Department of Computer Engineering , Maharashtra Institute of Technology, Pune.

Mrs. Suja Panickar
(Project Guide)

Dr.(Mrs.) V. Y. Kulkarni
(Head of Computer Engineering Department)

Place: Pune

Date:

ACKNOWLEDGEMENT

I take this opportunity to express my sincere appreciation for the cooperation given by Dr. (Mrs.) V. Y. Kulkarni, HOD (Department of Computer Engineering) and need a special mention for all the motivation and support.

I am deeply indebted to my guide Mrs. Suja Panickar for completion of this project report for which she has guided and helped me going out of the way.

For all efforts behind the project report, I would also like to express my sincere appreciation to staff of department of Computer Engineering, Maharashtra Institute of Technology Pune, for their extended help and suggestions at every stage.

Aseem Apastamb
Pushkar Ganorkar
Sarvesh Khire

Contents

List of Figures

List of Tables

Abstract

With the advent of deep learning the concept of end-to-end learning has been introduced; this has resulted in simplification of tasks in computer vision. As in traditional techniques, there was a need for multiple steps like feature extraction and required computer vision task, with deep learning this nullified the need for a pipeline of specialized and hand-crafted methods. In facial expression recognition systems deep neural networks have been used to automate extraction of intrinsic facial features along with classifying them into categories. There have been many studies in going on in FER which use complex deep learning models. We propose two different models which classify images and videos based on the facial expressions of human beings present. Along with that we show how it is comparable to the existing models.

Keywords: Computer Vision, Deep Learning, Deep Neural Networks, Facial expression recognition, feature extraction.

Chapter 1

Problem Definition

1.1 Problem Statement

The aim of the project is to perform facial expression recognition on the images that are provided by user using deep learning techniques. This task involves detecting the human faces present in the image and then identifying the underlying expression on the human face.

1.2 Problem Definition

Nowadays, Facial Expression Recognition has become a very important as it used in multi-disciplinary fields such as consumer mood analysis which is used by many companies for tracking the mood of consumer to help in personalized marketing. Other important applications include analysing attentiveness of car driver and analysing patient's emotional state. Hence, we intend to analyse existing state-of-the-art deep learning models for facial expression recognition and then develop and implement a deep neural network architecture for facial expression recognition. We also intend to draw comparisons between existing methods and our proposed methods.

Chapter 2

Introduction

Facial expressions play significant role in conveying the information which includes emotional states and intentions of human beings. The emotions are expressed in various forms like anger, fear, disgust, happiness, etc. Detecting and recognizing these emotions is crucial task for a human being. For a computer to recognize and understand an emotion is comparatively difficult than for a human being. There has been increasing interest in human emotion recognition in various fields including human computer interface, animation, medicine, security, etc. Human being expresses emotions in various ways like speech, face, text and body language. We are going to concentrate on facial expressions of a human.[10]

There are numerous methods in identifying a facial expression. Traditional methods are a two steps process which includes feature extraction and then classification. The feature extraction process was handcrafted. They included SIFT, HOG, LBP. Recently with latest deep learning techniques the extraction of the features has become simpler. In these latest techniques feature extraction and classification is done in a single step.

Although this seems simpler with the use of deep learning techniques, there exist many challenges to accomplish this task. Some of these challenges are low light, occlusions, multiple subjects, motion blur, translation invariance, etc. Also, the non-uniform nature of human face and facial pose and orientation conditions considerably affect the efficiency of the algorithms. Apart from this the data available is restrictive. Still there have been attempts made to overcome these problems.

Chapter 3

Recent research in Facial Expression Recognition

Convolutional Neural Networks have been leveraged for many applications involving images and videos. These tasks are image classification, object localization and detection, face recognition systems. In facial expression recognition systems too, various architectures based on Convolutional Neural Networks have been implemented. For example, AlexNet and GoogleNet have been used as a common feature extraction models in most of the implementations. The Facial expression recognition system generally classifies the facial images into 6 basic categories defined by Ekman. These classes are: anger, happiness, sadness, surprise, fear, disgust. There is a 7th class 'neutral' which has been added later. The facial expression is also determined by the Facial Action Units (FAUs). These FAUs are based on the movement of the facial muscles. Based on combination of these FAUs the facial expressions can be identified. The combination of different FAUs has resulted in creation of compound facial expression categories such sadly disgusted and happily surprised. Our goal is as far now is limited to classify into the 7 basic categories. [9][10]

For facial recognition system it is important to pre-process the images before actual training the model. Given a series of training data, the first step is to detect the face and then to remove background and non-face areas. Most implementations used Haar cascade face detector. Other implementations also use Viola-Jones face detector which is a classic and widely employed implementation for face detection, which is robust and computationally simple for detecting near-frontal faces. Although face detection is the only indispensable procedure to enable feature learning, further face alignment using the coordinates of localized landmarks can substantially enhance the FER performance. Recently, deep networks have been widely exploited for face alignment. Cascaded CNN is the early work which predicts landmarks in a cascaded way.[14]

Recent developments like attention mechanism have been explored in the context of facial expression recognition. With attention mechanism specific landmarks are focused upon to get the idea of underlying emotion [1, 3]. Another model used 3D CNN for extracting the temporal geometric data (landmarks) from the sequence of images, and the temporal appearance data was extracted using a CNN. A different approach dealt with the RGB and Depth map of the images and these were trained on two different neural network architecture in a joint fashion. The outputs of the two different neural networks were connected and facial expression classification was performed. In an approach using Multi-Task Cascaded Convolutional Neural Network, the standard dataset FER2013 was used for training the MTCNN. This is used for face alignment as well as facial expression recognition.[2, 5, 8, 14]

Chapter 4

Literature Survey

Sr. No.	Title	Publication	Concept	Gaps
1	Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network	ArXiv 2019	Use of Attention Mechanism by implementing spatial transformer.	The accuracy for FER2013 dataset remains 70% while that on others is very high. It seems the model finds it difficult to learn bigger dataset.
2	FERAtt: Facial Expression Recognition with Attention Net	IEEE CVPR 2019	Use of encoder-decoder Attention Mechanism in combination with feature extraction module.	Accuracy is very low for BU-3DFE dataset. There are 4 modules, each module is a neural network module, so there are huge number of parameters (weights) involved.
3	Frame Attention Networks For Facial Expression Recognition In Videos	IEEE ICIP 2019	Use of Attention Mechanism for expression recognition in videos.	Accuracy is extremely less on AFEW Dataset and not a huge increase from previous attempts.

Sr. No.	Title	Publication	Concept	Gaps
4	Video-Based Facial Expression Recognition Using a Deep Learning Approach	Springer Nature Singapore Ltd. 2019	OpenCV CAPPROP-POSMSEC for frame extraction. SSD ResNet10 for face detection and 5 layered ConvNet, 7 layered ConvNet	It is a very naive approach to detect facial expressions. The validation and testing accuracy is very low, as the model tends to overfit the training dataset.
5	Deep Facial Expression Recognition: A Survey	ArXiv 2019	This is a survey paper with details regarding various methods for facial expression recognition	-
6	Covariance Pooling for Facial Expression Recognition	ArXiv 2018	Inception-ResnetV1, SPDNet	In video-based FER, accuracy of only 32.5% is obtained, which is worse than the reported accuracy. This is because of relatively small size of AFEW dataset compared to parameters in the network.
7	Survey of Face Detection on Low-quality Images	IEEE AFGR 2018	1. Viola-Jones Haar AdaBoost 2. H0G-SVM 3. Faster R-CNN 4. S3FD	No new technique proposed, comparison of existing ones. Expression recognition not considered, just face detection.
8	Video Emotion Recognition With Concept Selection	IEEE ICME 2019	Linear SVM and RBF Kernel, along with kernel fusion.	New system to recognition emotion of whole video. Does not figure out expressions of specifically faces.

Sr. No.	Title	Publication	Concept	Gaps
9	Joint Fine-Tuning in Deep Neural Networks for Facial Expression Recognition	IEEE 2015	Deep Temporal Appearance-Geometry Network (DTAGN), Joint fine-tuning of 2 networks.	The algorithm confuses between anger and disgust under oulu-casia dataset. Involvement of three loss function might make training difficult. The algorithm was not trained on bigger datasets.
10	Action Unit Based Facial Expression Recognition Using Deep Learning	Springer 2017	CNN for calculating confidence values of active, not active for Action Units and SVRs to calculate values of arousal and valence.	-
11	Deep Learning Approaches for Facial Emotion Recognition: A Case Study on FER-2013	Springer 2017	AlexNet, GoogleNet.	The model is trained only on one dataset. The dataset is FER2013, and contains black and white images only.
12	Facial Expression Recognition via Joint Deep Learning of RGB-Depth Map Latent Representations	IEEE 2017	Jointly trains the neural networks on RGB images along with the depth map of the images present in BU-3DFE dataset.	Cannot be used for photos with less pixel density or black and white photos. The model is only trained on RGB images and their heat maps.

Sr. No.	Title	Publication	Concept	Gaps
13	Joint Face Detection and Facial Expression Recognition with MTCNN	IEEE ICISCE 2017	A new method for face detection and facial expression recognition using deep learning techniques is proposed.	Has low validation accuracy (60%). Slight overfitting of the data. No real time data was used.
14	Review and Comparison of Face Detection Algorithms	IEEE ICCCDSE 2017	Four basic algorithms are discussed and compared on the basis of precision and recall, calculated using DetEval software.	A comparative evaluation of techniques, new one not proposed. Only one sample image considered.
15	Style Aggregated Network for Facial Landmark Detection	IEEE CVF 2018	Style Aggregated Networks (SAN) - a combination of CycleGAN and Landmark prediction module.	-
16	Deep Temporal-Spatial Aggregation For Video-Based Facial Expression Recognition	IEEE 2019	AlexNet for both temporal and spatial streams.	Fear is identified with a accuracy of 50%, as this model is trained on limited and low-quality dataset. The accuracy of this model depends on large-scale and quality dataset. Design framework has a large number of parameters, and requires expensive computation.

Sr. No.	Title	Publication	Concept	Gaps
17	Spatio-Temporal Facial Expression Recognition Using Convolutional Neural Network And Conditional Random Fields	IEEE 2017	Inception-ResnetV4 for capturing spatial relations in facial images, CRF for capturing temporal relations between image frames.	During preprocessing, images cannot be resized to less than 299X299, as it affects recognition accuracy. In subject-independent case, there is no significant improvement over state-of-the-art techniques on CK+ and MMI dataset because the model is too deep for these databases (number of training examples are not high enough for proper training).
18	A Multi-Face Challenging Dataset for Robust Face Recognition	IEEE ICARCV 2018	Local Binary Pattern (LBP) and VGGFace CNN descriptor used to establish complexity of new dataset.	Just a study of standard datasets, and a new dataset proposed. Expression analysis techniques not considered, nor a new one proposed.
19	Facial Expression Recognition by Calculating Euclidian Distance for Eigen Faces using PCA	IEEE ICCSP 2018	Principal Component Analysis (PCA)	Performed on real time data, but no standard dataset considered. Uses PCA only. No variations in experimentation. Other techniques like CNN not used.

Sr. No.	Title	Publication	Concept	Gaps
20	DeXpression: Deep Convolutional Neural Network For Expression Recognition	ArXiv 2016	GoogleNet	-
21	Building a face expression recognizer and a face expression database for an intelligent tutoring system	IEEE ICALT 2017	Geometric based feature recognizer, with the help of Dlib library	The emotion Surprise is often confused with Disgust, at a rate of 0.045% on CK+ dataset. Initital images with no expressions are misclassified.
22	Recognition of Action Units in the Wild with Deep Nets and a New Global-Local Loss	IEEE ICCV 2017	Use of GL (global-local) Loss funtion on Convolutional Neural Network leads to fast convergence with significantly accurate results to determine 11 action units.	-

Sr. No.	Title	Publication	Concept	Gaps
23	Facial expression recognition using optimized active regions	Human-centric Computing and Information Sciences, Springer 2018	Divides the face into three active regions. Uses CNN for feature extraction and classifying expression in each active region, and selects the optimized active region based on majority voting.	The time to search optimized active regions is high. Shape of active region is considered to be square, but in reality shapes of mouth and eyes are somewhat rectangular. Facial landmarks are detected using API of Face++ company with limited accessibility unless there is internet. Accepts only grayscale images as input.

Table 4.1: Literature survey

1. **Paper 1**: DeXpression: Deep Convolutional Neural Network for Expression Recognition

   - **Authors**: Peter Burkert, Felix Trier, Muhammad Zeshan Afzal, Andreas Dengel and Marcus Liwicki
   - **Year of Publication**: 2016
   - **Summary**:
     The model proposed in this paper is a convolutional network based deep learning model. It is independent of any of the hand crafted feature extraction methods. The model is a complex one but has lesser number of parameters. The datsets used in the paper are CK+ and MMI. The model consists of two feature extraction blocks with parallel paths. The two paths contains differently sized filters which reflects the various scales at which faces are appearing.

2. **Paper 2**: FERAtt: Facial Expression Recognition with Attention Net

   - **Authors**: Pedro D. Marrero Fernandez, Fidel A. Guerrero Pena, Tsang Ing Ren

- **Year of Publication**: 2019
- **Summary**:
  The model used is divided into four parts, with each part has a specific task. The attention module is a deep encoder-decoder module with Residual networks. The input of extraction module and attention module is combined and transformed using a non-linear function and given to reconstruction module which reduces the size of image and enhances it. Then the representation and classification model takes this image as input and classifies into one of seven categories. The loss function presented serves the purpose of regularization. This approach has shown great results on CK+ (Cohn-Kanade extended) and BU-3DFE datasets. The model was also trained on the synthetic dataset generated using a synthetic generator.

3. **Paper 3**: Joint Fine-Tuning in Deep Neural Networks for Facial Expression Recognition

   - **Authors**: Heechul Jung Sihaeng Lee Junho Yim Sunjeong Park Junmo Kim
   - **Year of Publication**: 2015
   - **Summary**:
     In this research paper the temporal information was considered rather than a still image, which means the image sequence was considered as an input. They used two models one for extracting temporal appearance features and other for temporal geometry (landmarks). The two different networks were used for aforementioned two tasks and they were trained jointly producing single output. Interface algorithm was used for facial landmark detection. 3D CNN were used in capturing temporal changes in image sequences. The results of the joint model outperformed the state-of-the-art models.

4. **Paper 4**: Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network Recognition

   - **Authors**: Shervin Minaee, Amirali Abdolrashidi
   - **Year of Publication**:2019
   - **Summary**:
     The proposed model in the paper uses attention mechanism to classify the underlying emotion in the face images. The attention mechanism is implemented by using spatial transformer network. This model has shown great results, even with lesser number of layers (10) than existing models.

5. **Paper 5**: Facial Expression Recognition via Joint Deep Learning of RGB-Depth Map Latent Representations

   - **Authors**: Oyebade K. Oyedotun, Girum Demisse, Abd El Rahman Shabayek, Djamila Aouada, Bj¨orn Ottersten
   - **Year of Publication**:2017
   - **Summary**:
     The authors paper proposes to jointly train the neural networks on RGB images along with the depth map of the images present in BU-3DFE dataset. The authors posit that learning jointly would result in learning more discriminative features as against singly learning from either of the modalities.

6. **Paper 6**: Frame Attention Networks For Facial Expression Recognition In Videos

   - **Authors**: Debin Meng, Xiaojiang Peng, Kai Wang, Yu Qiao
   - **Year of Publication**:2019
   - **Summary**:
     This paper introduces a deep learning model for frame representation and expression classification. The FAN model consists of 2 parts 1. feature embedding module 2. frame attention module. The frame attention module learns two-level attention weights, i.e. self attention weights and relation-attention weights, which are used to adaptively aggregate the feature vectors to form a single discriminative video representation. The results show that the model achieved a great accuracy on CK+ dataset, but the AFEW 8.0 dataset challenged the model like it challenged other state-of-the art, still FAN out performed every other model in comparison.

7. **Paper 7**: Joint Face detection and Facial Expression Recognition with MTCNN

   - **Authors**: Jia Xiang, Gengming Zhu
   - **Year of Publication**: 2017
   - **Summary**:
     In this work, a new method for face detection and facial expression recognition using deep learning techniques is proposed. Specifically, the inherent correlation between them is calculated. An extensive set of experiments on the well-known FER2013 tested for FER work was conducted. Current accuracy is low.

8. **Paper 8**: Deep Temporal-Spatial Aggregation For Video-Based Facial Expression Recognition

   - **Authors**: Xianzhang Pan, Wenping Guo, Xiaoying Guo, Wenshu Li, Junjie Xu, Jinzhao Wu
   - **Year of Publication**:2019
   - **Summary**:
     The network proposed in the paper consist of 30 individual CNN streams(temporal CNN network and spatial CNN network). The temporal CNN network aims to generates temporal features from the optical flow signals and spatial CNN network aims to generate spatial features from facial images of the video.The network consist of a Aggregation layer called EmotionalVlan for aggregating the temporal and spatial features.The ouput of the Aggregation layer is given to the softmax layer for emotion recognition.This network achieves 46.1% accuracy on BAUM-1s database and 43.1% accuracy on eNTERFACE05 database.

9. **Paper 9**: Spatio-Temporal Facial Expression Recognition Using Convolutional Neural Network And Conditional Random Fields

   - **Authors**: Behzad Hasani, Mohammad H. Mahoor
   - **Year of Publication**:2017
   - **Summary**:
     This paper proposes a two part network for the facial expression recognition in videos. The first part is a DNN based architecture consisting of Convolutional layer followed by three Inception-Resnet layers and two fully connected layers. This part of the network helps us in extracting the spatial relations within the frames. The second part of the network is a linear Conditional Random Fields model which helps us in extracting temporal relations between the frames. This network is tested in two different cases subject-independent and crossDatabase on three datasets CK+,MMI and FERA. Results show that the proposed network outperforms the current state of the art techniques in crossDatabase case and gives comparable performance in subject-independent case.

Chapter 5

Concepts Required for the Project

5.1 Deep Learning

Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher level features from the raw input. For example, in image processing, lower layers may identify edges, while higher layers may identify the concepts relevant to a human such as digits or letters or faces.
Deep learning architectures such as deep neural networks, deep belief networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases superior to human experts.

5.2 Deep Neural Network

A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers. The DNN finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a non-linear relationship. DNNs can model complex non-linear relationships. DNN architectures generate compositional models where the object is expressed as a layered composition of primitives. The extra layers enable composition of features from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network. Deep architectures include many variants of a few basic approaches. Each architecture has found success in specific domains. It is not always possible to compare the performance of multiple architectures, unless they have been evaluated on the same data sets.Recurrent neural networks (RNNs), in which data can flow in any direction, are used for applications such as language modeling. Long short-term

memory is particularly effective for this use. Convolutional deep neural networks (CNNs) are used in computer vision.

5.3 Convolutional Neural Network

The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers. A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of a series of convolutional layers that convolve with a multiplication or other dot product. The activation function is commonly a RELU layer, and is subsequently followed by additional convolutions such as pooling layers, fully connected layers and normalization layers, referred to as hidden layers because their inputs and outputs are masked by the activation function and final convolution. The final convolution, in turn, often involves backpropagation in order to more accurately weight the end product. Below are the terminologies in Convolutional Neural Networks:

5.3.1 Convolution

When programming a CNN, each convolutional layer within a neural network should have the following attributes:

- Input is a tensor with shape (number of images) x (image width) x (image height) x (image depth).
- Convolutional kernels whose width and height are hyper-parameters, and whose depth must be equal to that of the image. Convolutional layers convolve the input and pass its result to the next layer. This is similar to the response of a neuron in the visual cortex to a specific stimulus.

5.3.2 Pooling

Convolutional networks may include local or global pooling layers to streamline the underlying computation. Pooling layers reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer.

5.3.3 Fully-connected

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.

5.3.4 Receptive Field

In neural networks, each neuron receives input from some number of locations in the previous layer. In a fully connected layer, each neuron receives input from every element of the previous layer. In a convolutional layer, neurons receive input from only a restricted subarea of the previous layer. The input area of a neuron is called its receptive field. So, in a fully connected layer, the receptive field is the entire previous layer. In a convolutional layer, the receptive area is smaller than the entire previous layer.

5.3.5 Weights

Each neuron in a neural network computes an output value by applying a specific function to the input values coming from the receptive field in the previous layer. The function that is applied to the input values is determined by a vector of weights and a bias (typically real numbers). Learning, in a neural network, progresses by making iterative adjustments to these biases and weights. The vector of weights and the bias are called filters and represent particular features of the input.

5.4 Various CNN architectures

5.4.1 VGGNet

VGG Net is a plain and straight forward CNN architecture among all other. Thought it looks simple, it do outperform many complex architectures. It is the 1st runner-up in ImageNet Challenge in 2014. As shown above, there are totally 6 VGGNet Architectures. Among them, VGG-16 and VGG-19 are popular. The idea of VGG architectures is quite simple. we have to stack the convolutional layers with increasing filter sizes. i.e., if layer-1 has 16 filters, then layer-2 must have 16 or more filters. Another noteworthy point is that in every VGG architecture, all filters are of size 3*3. The idea here is that, two 3*3 filters almost cover the are of what a 5*5 filter would cover and also two 3*3 filters are cheaper than one 5*5 filter(cheaper in the sense of total no. of multiplications to be performed).

5.4.2 InceptionNet

GoogleNet team was the winner of the 2014-ILSVRC competition and is also known as Inception-V1. The main objective of the InceptionNet architecture was to achieve high accuracy with a reduced computational cost. It introduced the new concept of inception block in CNN, whereby it incorporates multi-scale convolutional transformations using split, transform, and merge idea. This block encapsulates filters of different sizes (1x1, 3x3, and 5x5) to capture spatial information at different scales (both at fine and coarse grain level). The exploitation of the idea of split, transform, and merge by GoogleNet, helped in addressing a problem related to the learning of diverse types of variations present in the same category of different images.

5.4.3 ResNet

A residual neural network (ResNet) is an artificial neural network (ANN). Residual neural networks do this by utilizing skip connections, or shortcuts to jump over some layers. Typical ResNet models are implemented with double- or triple- layer skips that contain nonlinearities (ReLU) and batch normalization in between. One motivation for skipping over layers is to avoid the problem of vanishing gradients, by reusing activations from a previous layer until the adjacent layer learns its weights.

Chapter 6

Mathematical Model

6.1 Basic Mathematical Model

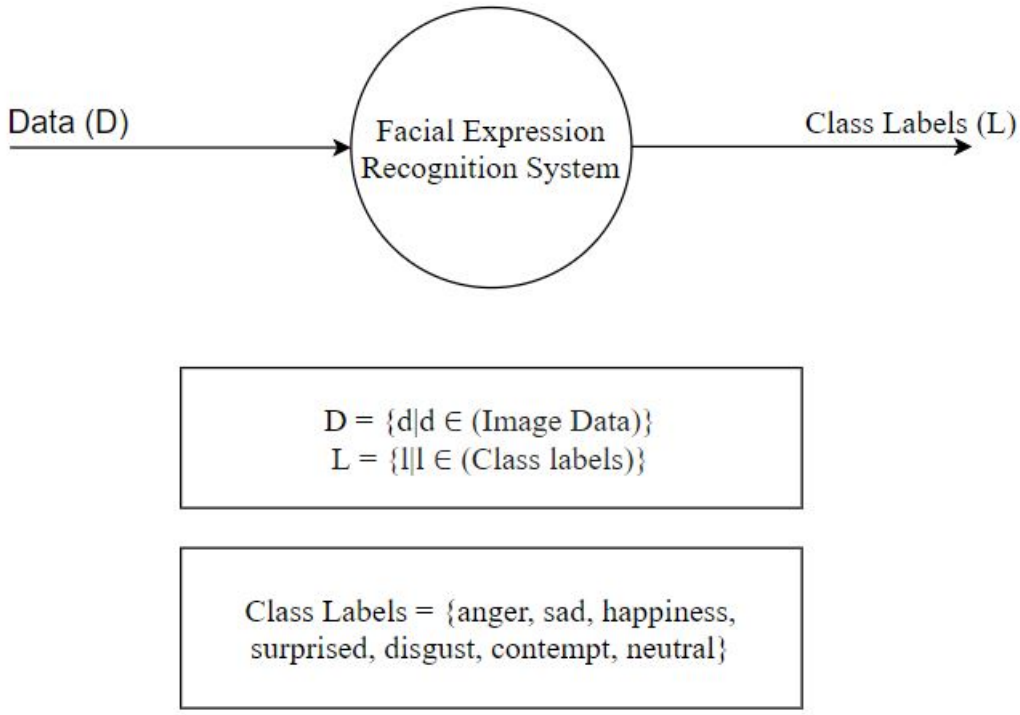

Figure 6.1: Basic Mathematical Model

6.2 Face Detection Module

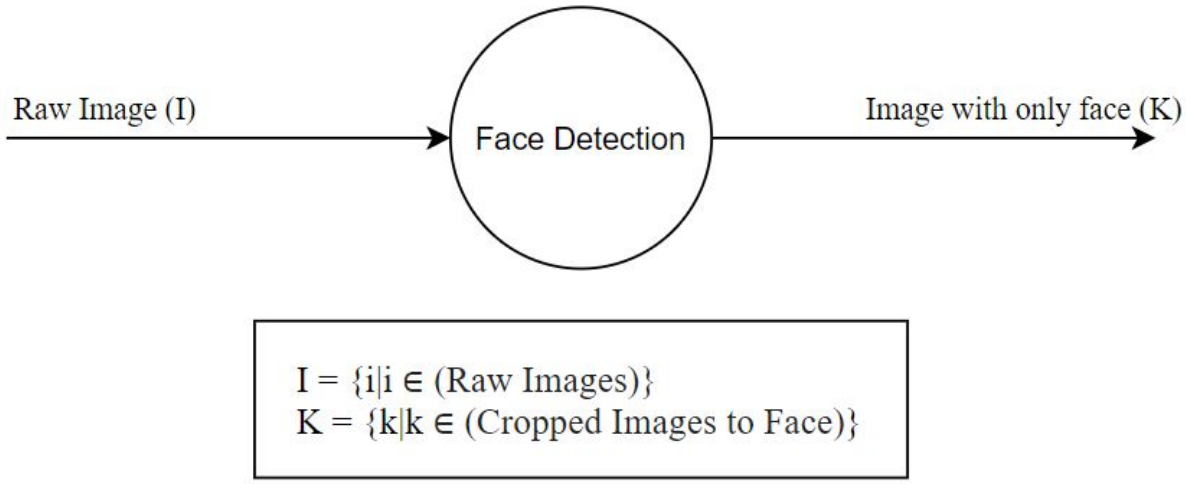

Figure 6.2: Face Detection Module

6.3 Data Augmentation Module

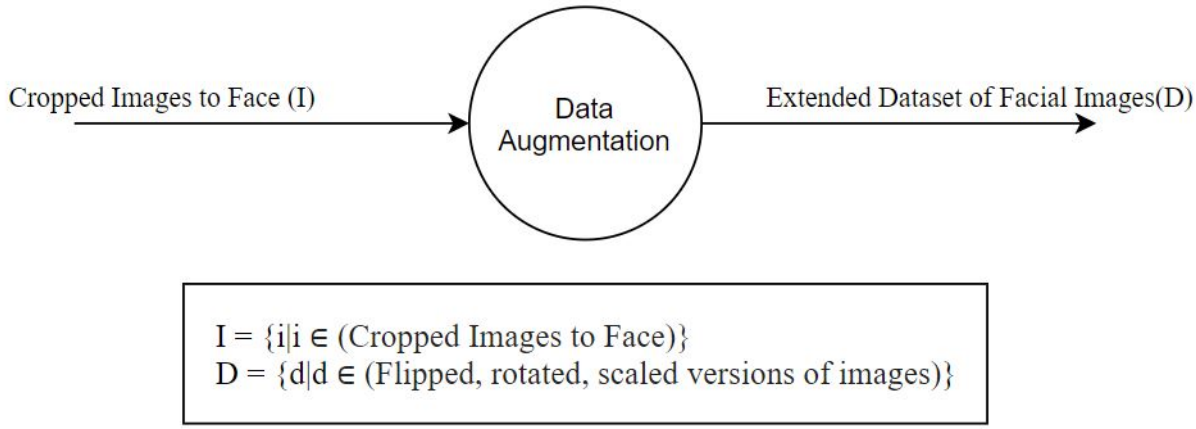

Figure 6.3: Data Augmentation Module

6.4 Feature Extraction Module

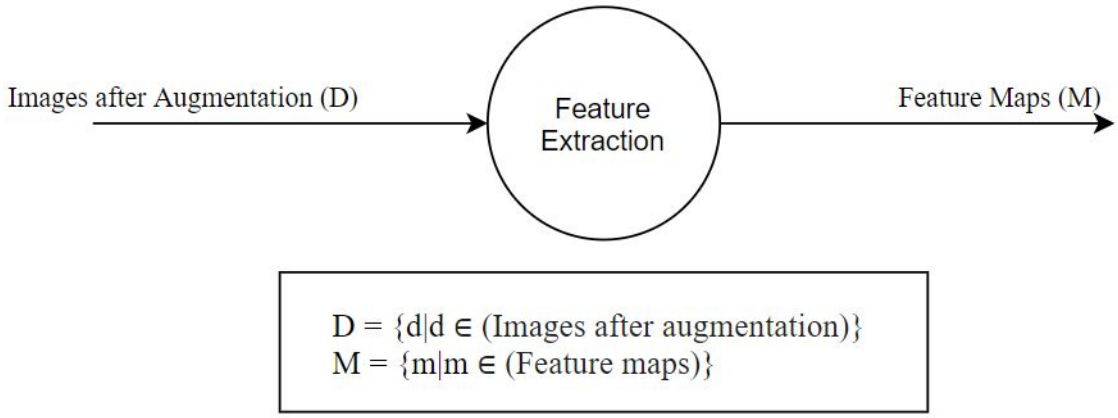

Figure 6.4: Feature Extraction Module

6.5 Classification Module

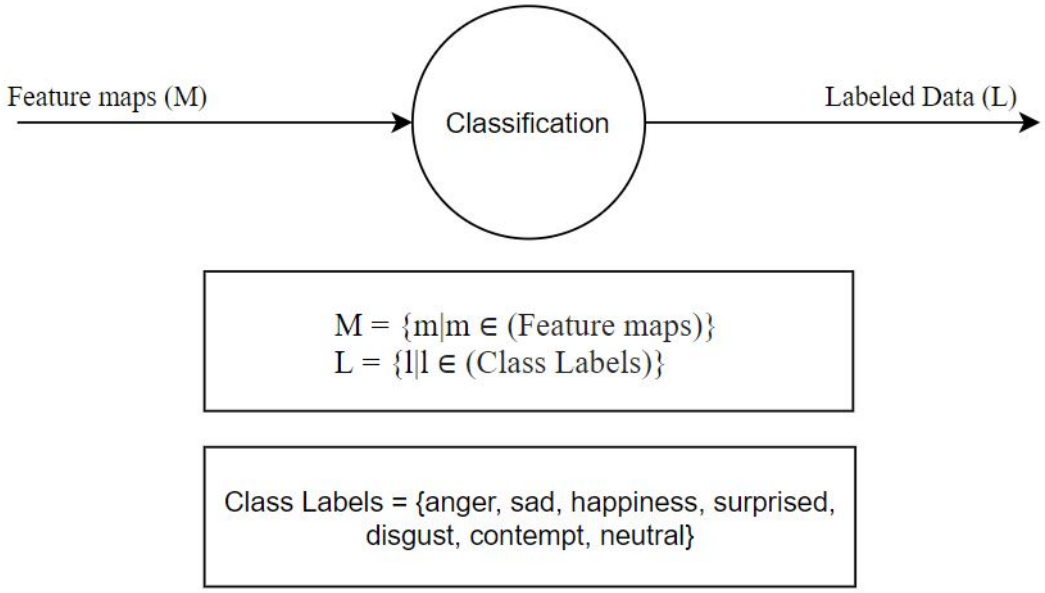

Figure 6.5: Classification Module

Chapter 7

Standard Datasets

It is important for facial expression recognition systems to have enough labeled training data with all possible variations. In this section we will explore publicly available datasets that contain basic expressions.[10, 11]

7.1 FER2013

The FER2013 [24] database was introduced during the ICML 2013 Challenges in Representation Learning. FER2013 is a large-scale and unconstrained database collected automatically by the Google image search API. All images have been registered and resized to 48*48 pixels after rejecting wrongfully labeled frames and adjusting the cropped region. FER2013 contains 28,709 training images, 3,589 validation images and 3,589 test images with seven expression labels (anger, disgust, fear, happiness, sadness, surprise and neutral).

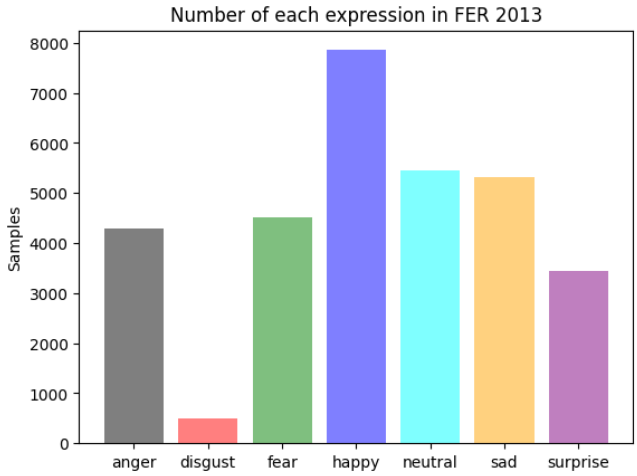

Figure 7.1: Distribution of the Classes in FER 2013

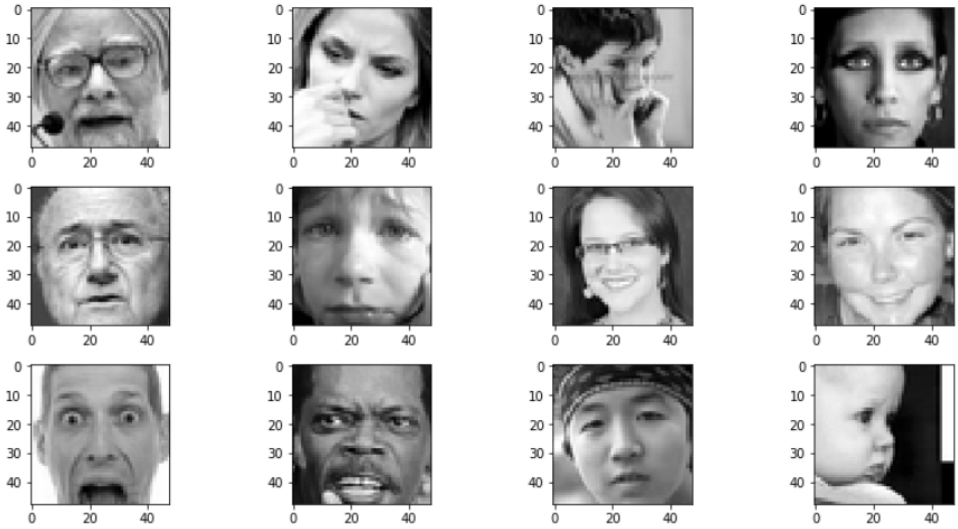

Figure 7.2: FER2013 Sample Images [24]

7.2 KDEF

KDEF [27] is a short form for Karolina Directed Emotional Faces. This dataset consists of 4900 pictures of human facial expressions. The set of pictures contains 70 individuals displaying 7 different emotional expressions. Each expression is viewed from 5 different angles. These 7 emotions are Anger, Happiness, Sadness, Disgust, Surprise, Fear, and Neutral.

The Subjects in the dataset were as follows:

- Population: 70 amateur actors, 35 females and 35 males.
- Selection criteria: Age between 20 and 30 years of age. No beards, moustaches, earrings or eyeglasses, and preferably no visible make-up during photo-session.

In this project we included images of people which were facing straight into the camera. So, the dataset that we used was reduced to 949 images. This was divided into train-test-validation into a ratio 70:15:15. The images were in RGB format initially, we used a grayscale format of the image for training the model. Even the size of the image in the dataset was fixed to 200 x 200 pixels. The distribution graph describes the data distribution of only the front facing images in the dataset

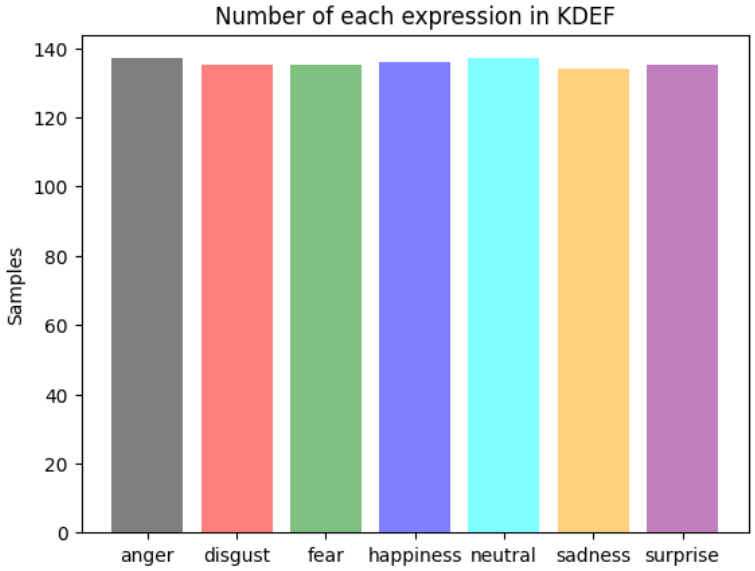

Figure 7.3: Distribution of the Classes in KDEF

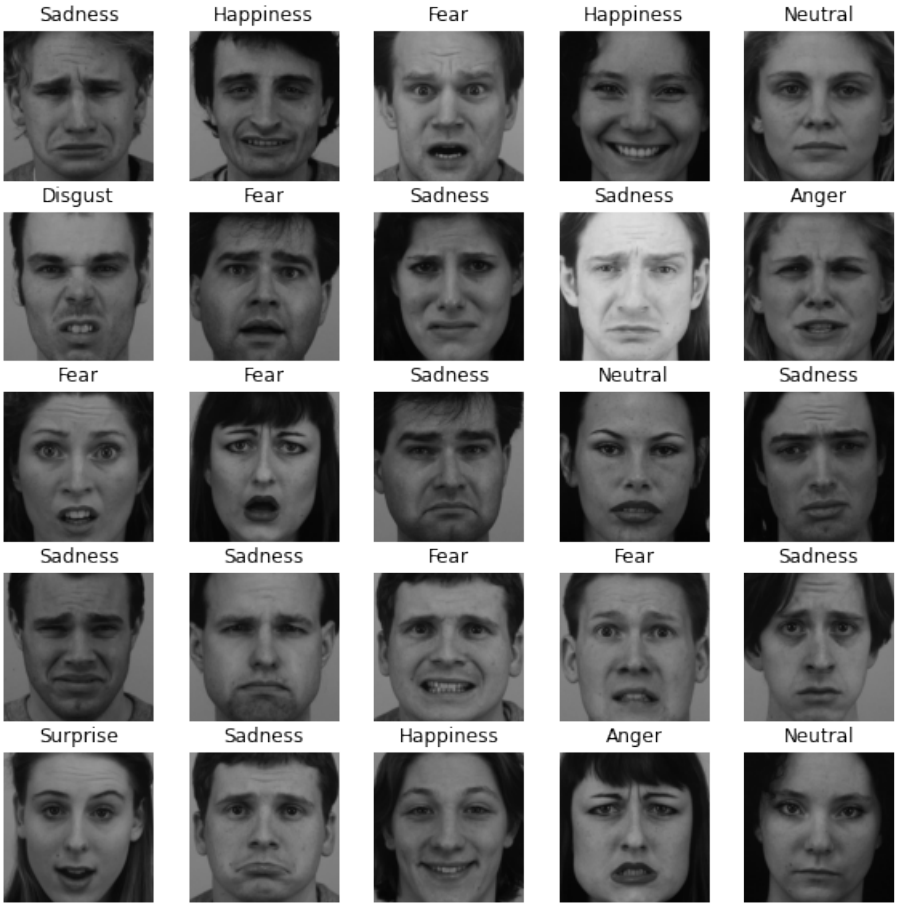

Figure 7.4: KDEF Sample Images

7.3 CK+

The Extended CohnKanade (CK+) [25]database is the most extensively used laboratory-controlled database for evaluating FER systems. CK+ contains 593 video sequences from 123 subjects. The sequences vary in duration from 10 to 60 frames and show a shift from a neutral facial expression to the peak expression. Among these videos, 327 sequences from 118 subjects are labeled with seven basic expression labels (anger, contempt, disgust, fear, happiness, sadness, and surprise) based on the Facial Action Coding System (FACS). Because CK+ does not provide specified training, validation and test sets, the algorithms evaluated on this database are not uniform.

The bar graph below gives the distribution of the images present in the dataset after filtering the dataset by removing the lookalike neutral images.

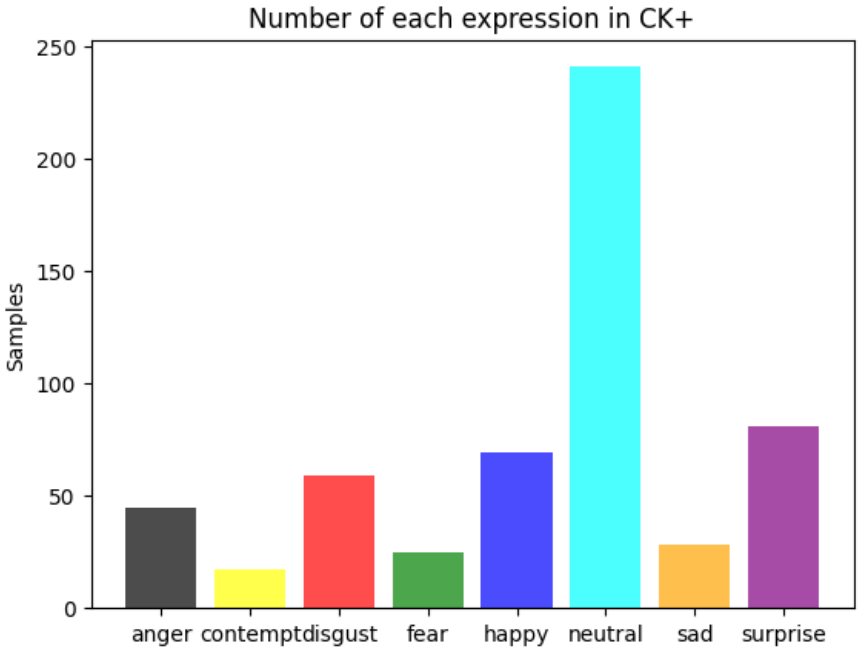

Figure 7.5: Distribution of the Classes in CK+

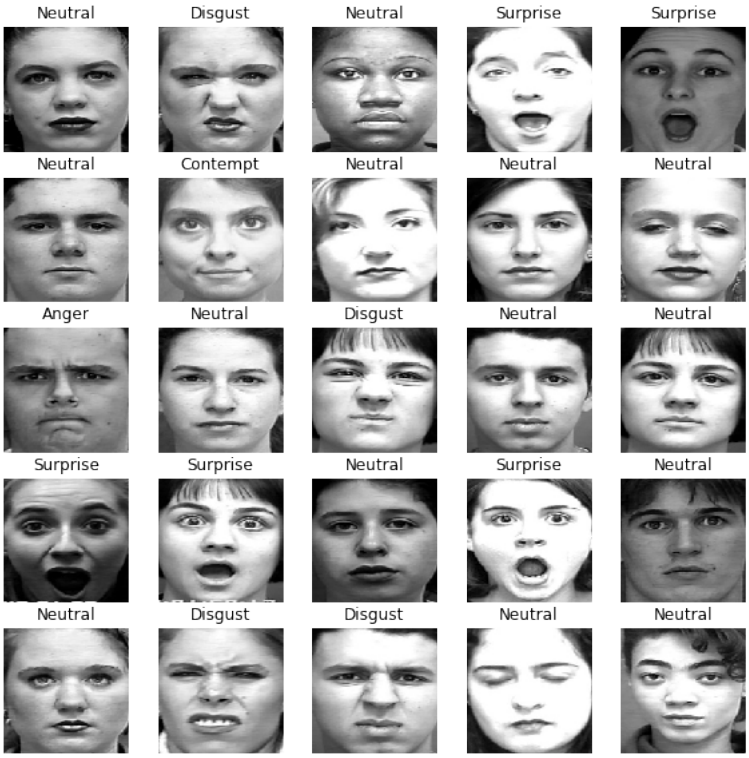

Figure 7.6: CK+ Sample Images

7.4 JAFFE

The Japanese Female Facial Expression (JAFFE) [26] database is a laboratory-controlled image database that contains 213 samples of posed expressions from 10 Japanese females. Each person has 3˜4 images with each of six basic facial expressions (anger, disgust, fear, happiness, sadness, and surprise) and one image with a neutral expression. The database is challenging because it contains few examples per subject/expression. Typically, all the images are used for the leave-one-subject-out experiment.

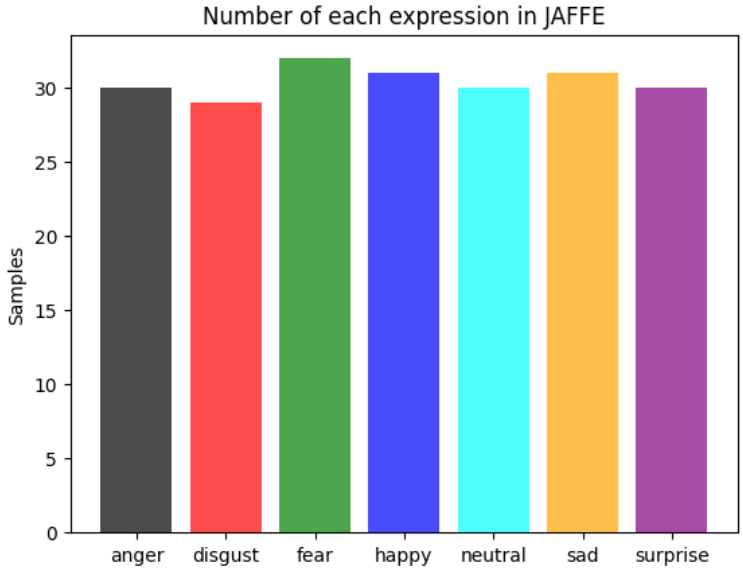

Figure 7.7: Distribution of classes in JAFFE

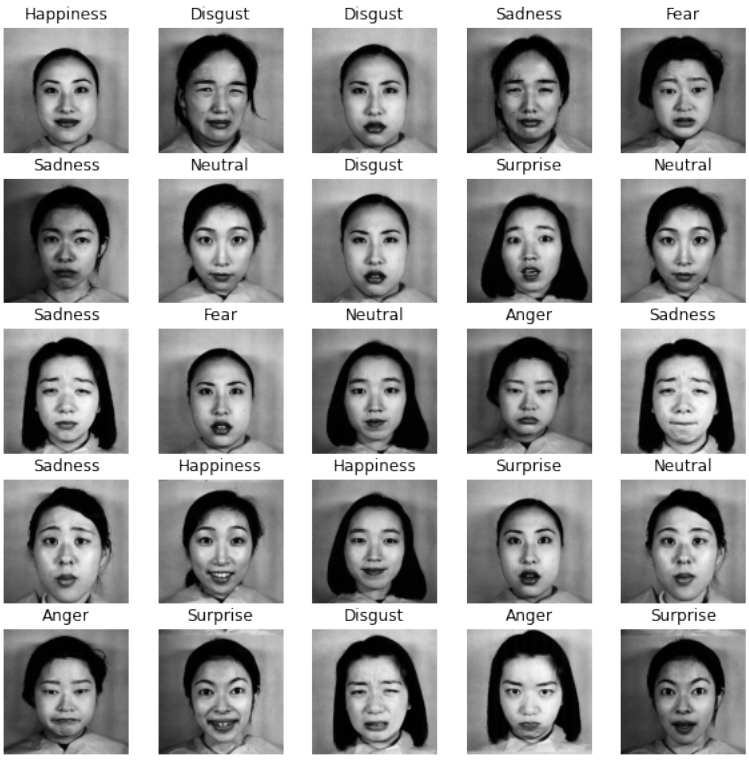

Figure 7.8: JAFFE Sample Images

7.5 SFEW

The Static Facial Expressions in the Wild (SFEW) [29] was created by selecting static frames from the AFEW database by computing key frames based on facial point clustering. The most commonly used version, SFEW 2.0, was the benchmarking data for the SReco sub-challenge in EmotiW 2015. SFEW 2.0 has been divided into three sets: Train (958 samples), Val (436 samples) and Test (372 samples). Each of the images is assigned to one of seven expression categories, i.e., anger, disgust, fear, neutral, happiness, sadness, and surprise. The expression labels of the training and validation sets are publicly available, whereas those of the testing set are held back by the challenge organizer.

7.6 AFEW

The Acted Facial Expressions in the Wild (AFEW) [28] database has served as an evaluation platform for the annual Emotion Recognition In The Wild Challenge (EmotiW) since 2013. AFEW contains video clips collected from different movies with spontaneous expressions, various head poses, occlusions and illuminations. AFEW is a temporal and multimodal database that provides with vastly different environmental conditions in both audio and video. Samples are labeled with seven expressions: anger, disgust, fear, happiness, sadness, surprise and neutral. The annotation of expressions have been continuously updated, and reality TV show data have been continuously added. The AFEW 7.0 in EmotiW 2017 is divided into three data partitions in an independent manner in terms of subject and movie/TV source: Train (773 samples), Val (383 samples) and Test (653 samples), which ensures data in the three sets belong to mutually exclusive movies and actors.

Chapter 8

Software Requirement Specification

8.1 Purpose

The purpose of this project is to provide a system that will take images and videos as input from the user, and accurately detect the human faces. Following this, the system will identify the expressions and classify it into one of the following basic categories: anger, sadness, happiness, fear, disgust, surprise, neutral. The first six classes were defined by Ekman, and the last one was an extension that was added later.

8.2 User class and Characteristics

In this system there are 2 entities:

- **User Application**: This is a user level application to connect the user with the system.
- **Underlying System**: This is the main module of the system, it will be responsible for classifying the expressions in images that the user inputs.

8.3 System Overview

Below is the block diagram for complete System.

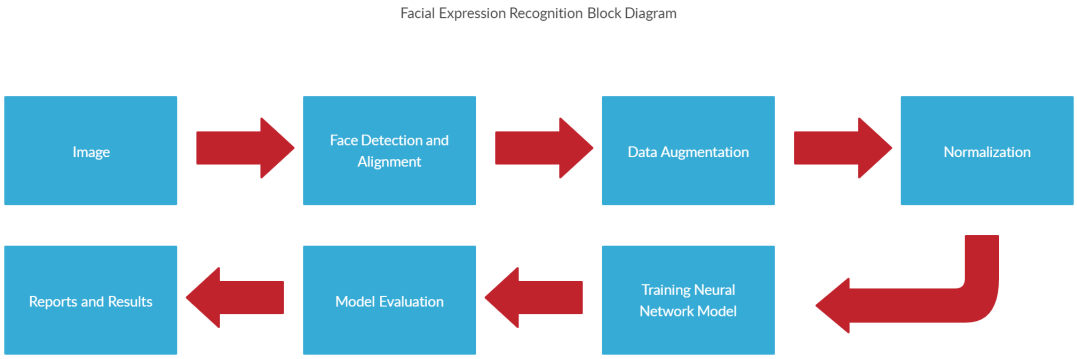

Figure 8.1: Block Diagram

8.4 Functional Requirements

- System must accept input in the form of images and videos.
- The images can be in grayscale format or coloured format.
- The images should have human faces, and they should be validated.
- System should be able to detect the faces and identify the expressions.
- The identified expression should be classified into one of the following categories: anger, sadness, happiness, surprise, fear, disgust, neutral.

8.5 Non-Functional Requirements

- The system should work efficiently and smoothly.
- The system should be able to handle huge datasets.
- The system should not overfit or underfit the training data.
- The system should accurately detect the faces and classify into the above categories.

8.6 Hardware Requirements

- Laptop or Desktop with i5 processor
- minimum 8 GB RAM and 100 GB Free storage
- Graphical Processing Unit (GPU) native or online

8.7 Software Requirements

- OS Windows 10 or Ubuntu 18.04 or LTS version
- Python 3.6+
- Deep learning libraries like Tensorflow 2.0, Pytorch, etc.
- Other Computer Vision libraries like OpenCV.

Chapter 9

UML Diagrams

9.1 Use-Case Diagram

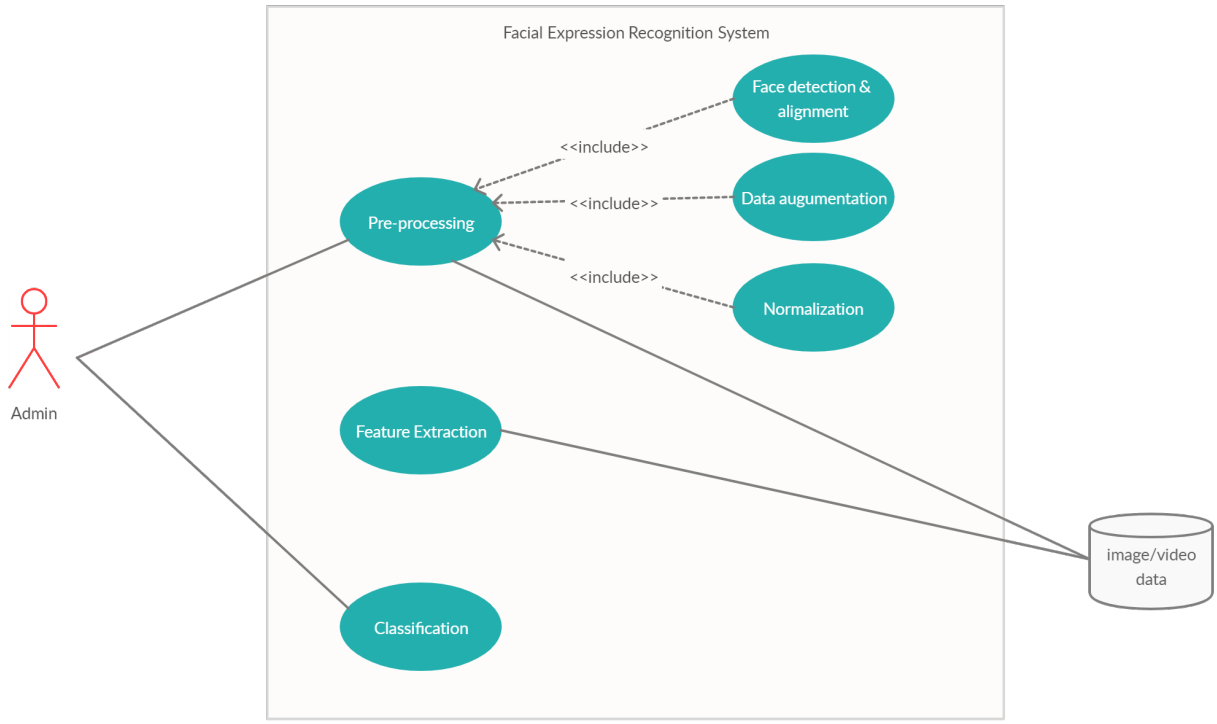

Figure 9.1: Use case diagram

9.2 Activity Diagram

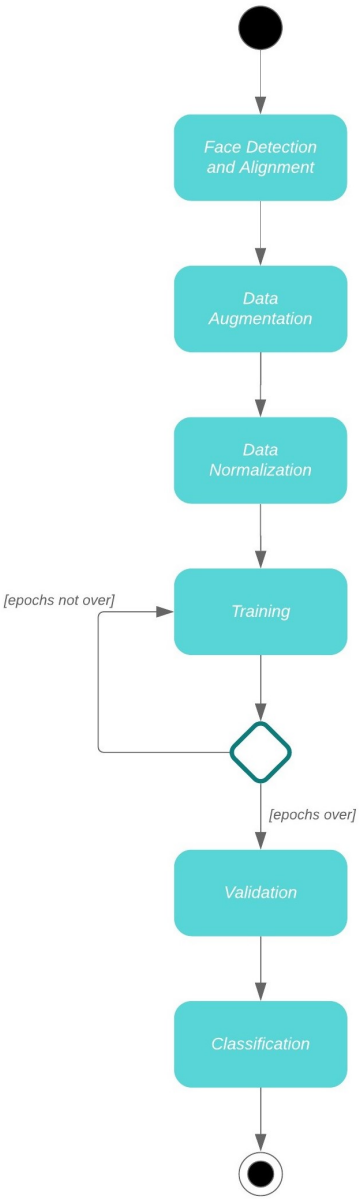

Figure 9.2: Activity diagram

9.3 Sequence Diagram

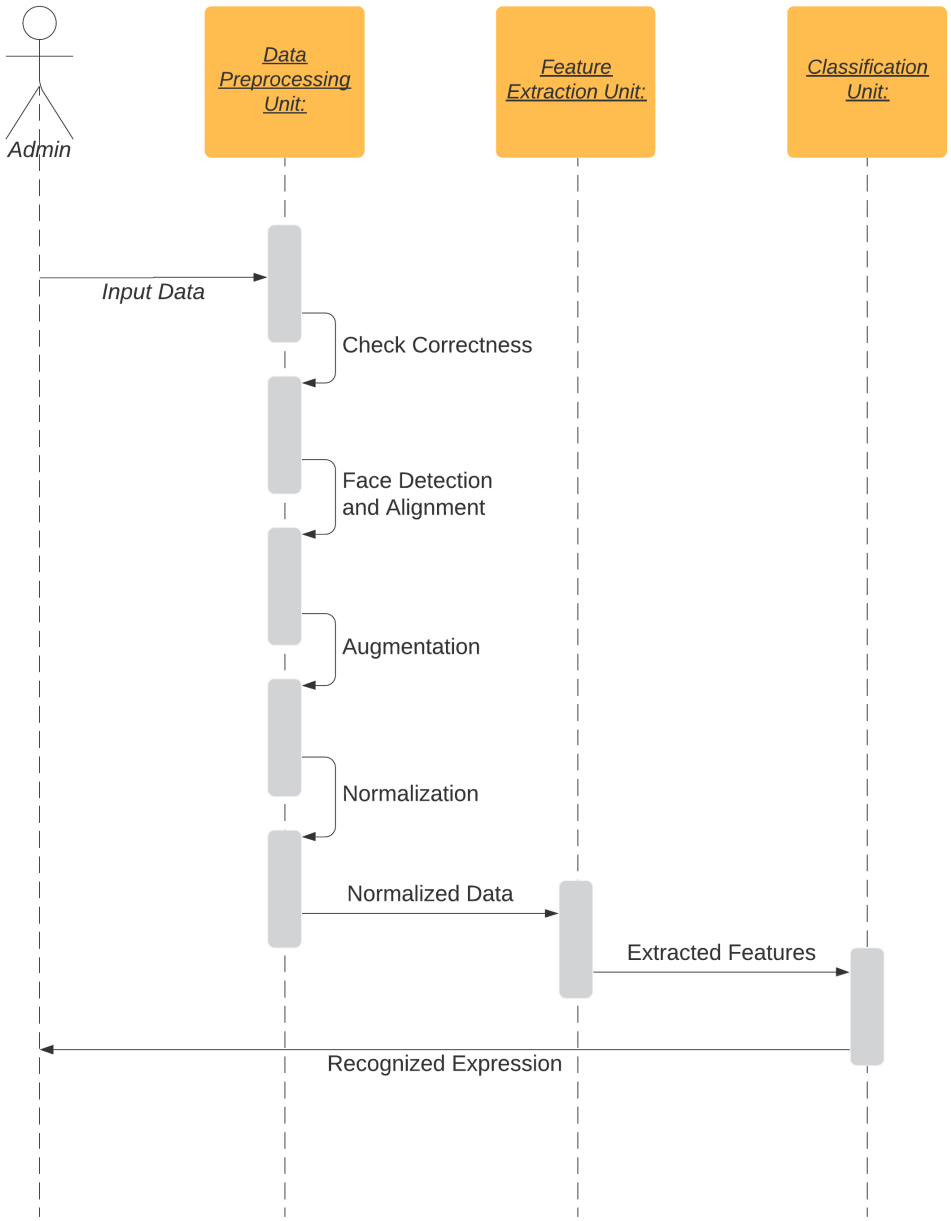

Figure 9.3: Sequence diagram

9.4 Data Flow Diagrams

9.4.1 Level - 0

Figure 9.4: Level - 0

9.4.2 Level - 1

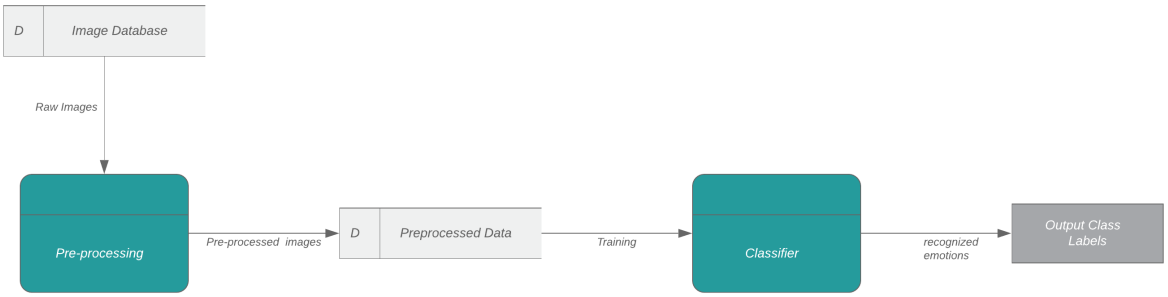

Figure 9.5: Level - 1

Chapter 10

Proposed Work

This column of the report demonstrates ideas that we intend to implement. It also contains models that are currently state-of-the-art.

The most basic model consists:

10.1 Pre-processing module

Variations that are irrelevant to facial expressions, such as different backgrounds, illuminations and head poses, are fairly common in unconstrained scenarios. Therefore, before training the deep neural network to learn meaningful features, pre-processing is required to align and normalize the visual semantic information conveyed by the face. Various steps involved are:

1. Face alignment and detection - Haar Cascade Frontal Face Detector or Multi-Task Cascade Convoluted Neural Network (MTCNN)[14]
2. Face data augmentation
3. Face normalization

10.2 Feature extraction

The CNN architecture is used to extract the features from a facial image. This CNN or the extractor can be existing architectures like InceptionNet [30], AlexNet [34] , ResNet [31], VGG [32] or a newly designed architecture. The different architectures vary based on the configuration of the CNN. In addition to this, these architectures can also be modified by making changes to the furthermost layers.

10.3 Output Layer

This is used for classifying the different emotions based on the features extracted above. Since we are doing multiclass classification, the prominent function that can be used is SoftMax.

In mathematics, the softmax function, also known as softargmax or normalized exponential function, is a function that takes as input a vector of K real numbers, and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers.

The Softmax function is given as

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \text{ for i=1,...,K and z} = (z_1, ...z_2) \; \epsilon \; \mathbb{R}^K \tag{10.1}$$

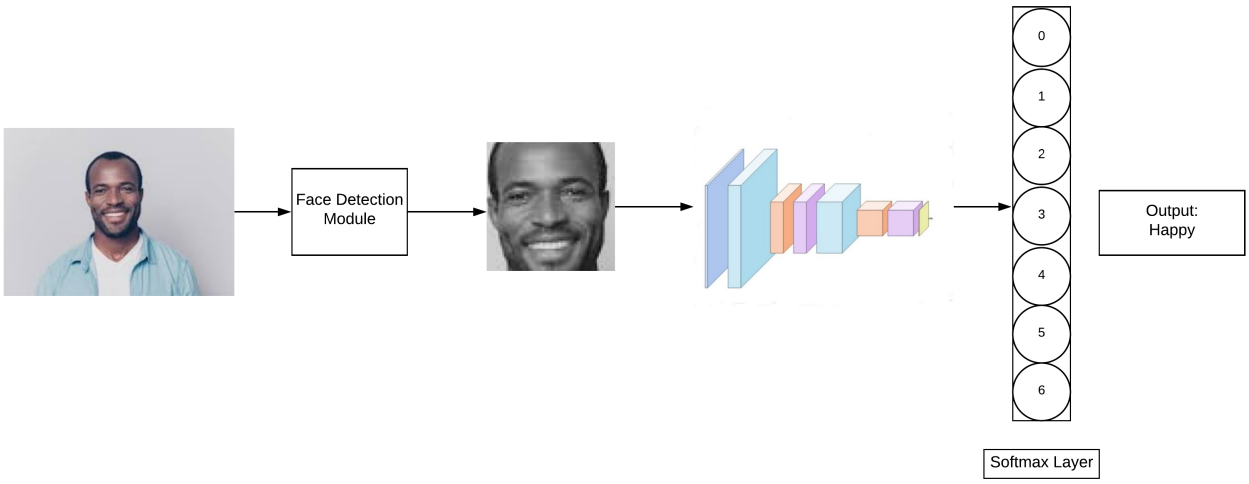

Figure 10.1: System Architecture

The above diagram depicts the architecture that we are going to use in our facial expression recognition system. The image of a person with is fed as an input to face detection module for detecting the face. The facial image is then fed to the feature extractor CNN for feature extraction process. Then the output expression is obtained using the softmax function. The feature extractor can be any standard CNN architecture like AlexNet [34], InceptionNet [30] and ResNet50 [31] or a custom built CNN architecture.

Chapter 11

Project Implementation

11.1 Overview

In this project we have implemented different deep learning models based on Convolutional Neural Networks (CNN) on different datasets which are described in the previous chapters. Data pre-processing is done on each dataset according to the requirements of each model. This includes face detection, resizing and reshaping, grayscaling, and augmentation. The image from the dataset can be in any format like RGB or grayscale. One has to decide what kind of format he has to use for training the model. Every image in the dataset should be in the same format. We have considered grayscale format for training. Additionally, one also has to keep each image in one particular size before training process. For some datasets we have stored only the face in the image (cropped) — after detecting the face in the image. In the following sections each model is described in detail with the input it requires and output it generates.

11.2 Tools and Technologies

The entire project is implemented using the Python 3.6+ language. The Deep learning models were implemented using the Keras API in Tensorflow 2.x. The image data preprocessing was done using the OpenCV library in Python. The data augmentation and callbacks were implemented using the Keras API functions.

For the implementation of this project two important programming environments were used: Anaconda Jupyter Notebooks and Google Colab which provides online free GPU (Nvdia Tesla K80). The deployment was done using the FLASK API in python. The web page was designed using html and server side using python.

11.3 Algorithm Details: Deep learning Architectures

The Deep Learning architectures that we considered for implementation are inspired from widely known CNN architectures like VGG-16 , ResNet, InceptionNet, DeXpression and Xception. These are fundamental architectures which have achieved good results on popular datasets. We implemented our models after researching these architectures. These models differ with each other in aspects like: the number of parameters, filters and filter sizes for both convolutional and pooling layer, input and output layers and many more. Every model is a combination of convolutional layer, max pooling layer, batch normalization layer, dropout layer and fully connected layer. These are the fundamental blocks of every model.

11.3.1 Models trained using KDEF dataset

We selected the images of people who were facing straight towards the camera. The total number of images then reduced to 949. Before splitting the images into training and validation set, they were pre-processed. For every image we used haarcascade frontal face detector to extract the facial part of the image. After face detection, we used a common format for all images throughout the dataset i.e. each image was a grayscale with size 200 x 200. The following are the models that were trained on the KDEF dataset:

DeXpression[22]

The first model that we used is the DeXpression model [22]. This model was introduced in the paper DeXpression:Deep Convolutional Neural Network for Expression recognition in the year 2016. We made slight changes in the implementation of the model .The difference between the actual model and the model we implemented is the change in the input shape. We implemented the DeXpression model using the Keras API in TensorFlow. The actual model described in the paper was trained on CK+ and MMI datasets, whereas we trained this model on KDEF dataset.

The DeXpression model is divided into four parts:

1. First part contains convolutional layers with 64 (7 x 7) filters, with activation function Relu, max-pooling layer and layer normalization. This is used for image pre-processing.

2. The second and third parts are the parallel feature extraction parts. These consists of convolutional, pooling and Relu layers. The parallel feature extraction

blocks have two paths. In the first path the features are down sampled and then the filter is applied and in the other path filter is directly applied to the input features. The output of the two paths is concatenated which acts as an input for the next parallel feature extraction block. There are two such parallel feature extraction blocks.

3. The last part contains a fully connected layer which is obtained on flattening the output of the convolutional layers. The last layer is the softmax layer containing 7 units, which is used for classification of expressions.

The dropout layer was added in between the fully connected layers to reduce overfitting. With addition of dropout layer we could deactivate some random neurons while training, which in turn helped to generalise the model. The dropout rate was 25 percent.

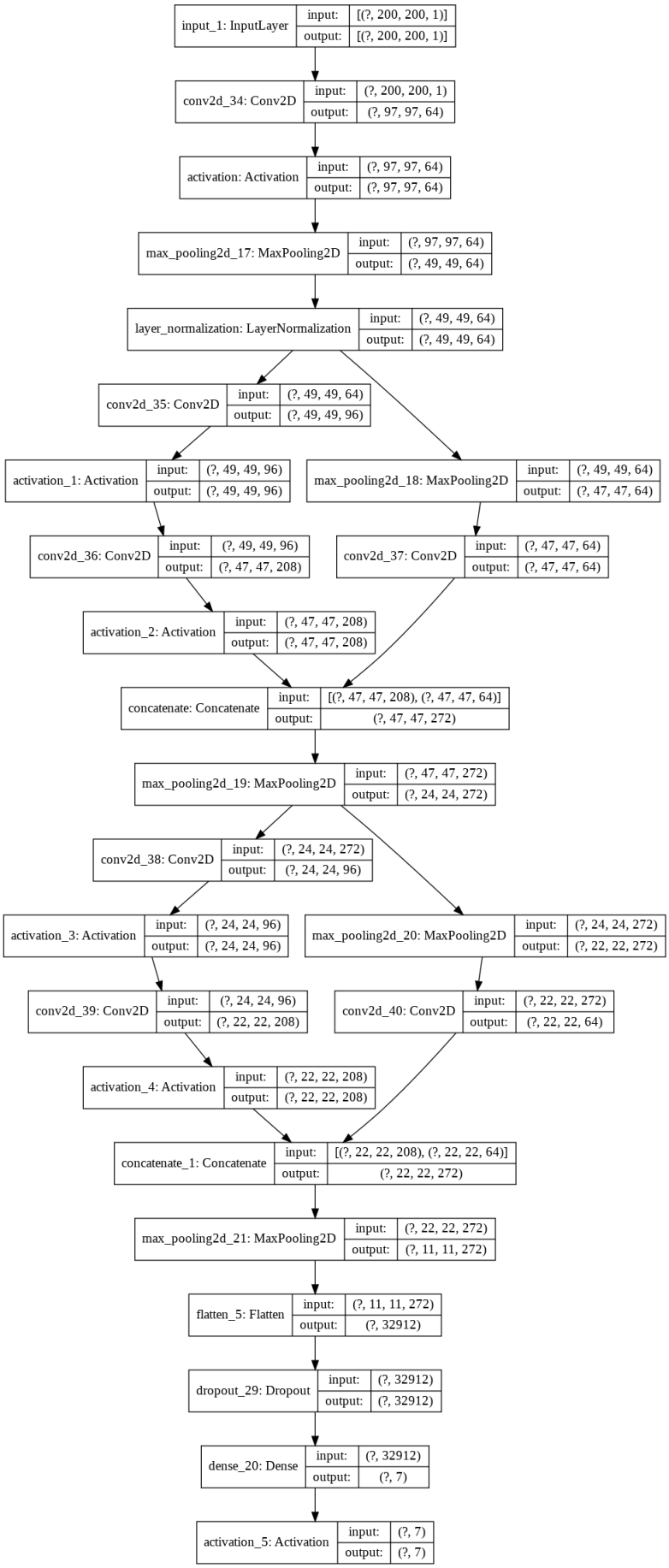

Figure 11.1: DeXpression Model Architecture

Model 2: simple model 1

This model is also somewhat inspired by the structure of the InceptionNet model. The first step was to preprocess the dataset. We converted the images into single-channel black and white images. Then using haar cascade we cropped the faces from the original image, and then resized it into 200 x 200 pixels. The model accepts these 200 x 200 images as input.

Initially, a 7 x 7 filter was used to extract the features, followed by pooling and layer normalization. Then the model splits into two paths: the first path consists of three convolutional layers, while the second path consists of two such layers. The first path has convolutional layers with a 5 x 5, 3 x 3 and 1 x 1 filter, while second path does not contain 1 x 1 filters. The 1 x 1 filter reduces the dimensionality which decreases the number of feature maps while retaining their salient features. Then the outputs of these two paths are concatenated. This layer feeds into a layer which flattens it into a single dimensional vector. Finally the fully connected layer accepts this vector as input and gives the final prediction.

Similar to the Dexpression model, we also used the Layer Normalization after the first convolutional layer to ensure exactly same number of computations are performed at training and testing size. It normalizes the activation of previous layer for every sample in a batch independently.

Max pooling with stride 2 and filter 3 is used to downsample an input representation by highlighting the most present features.

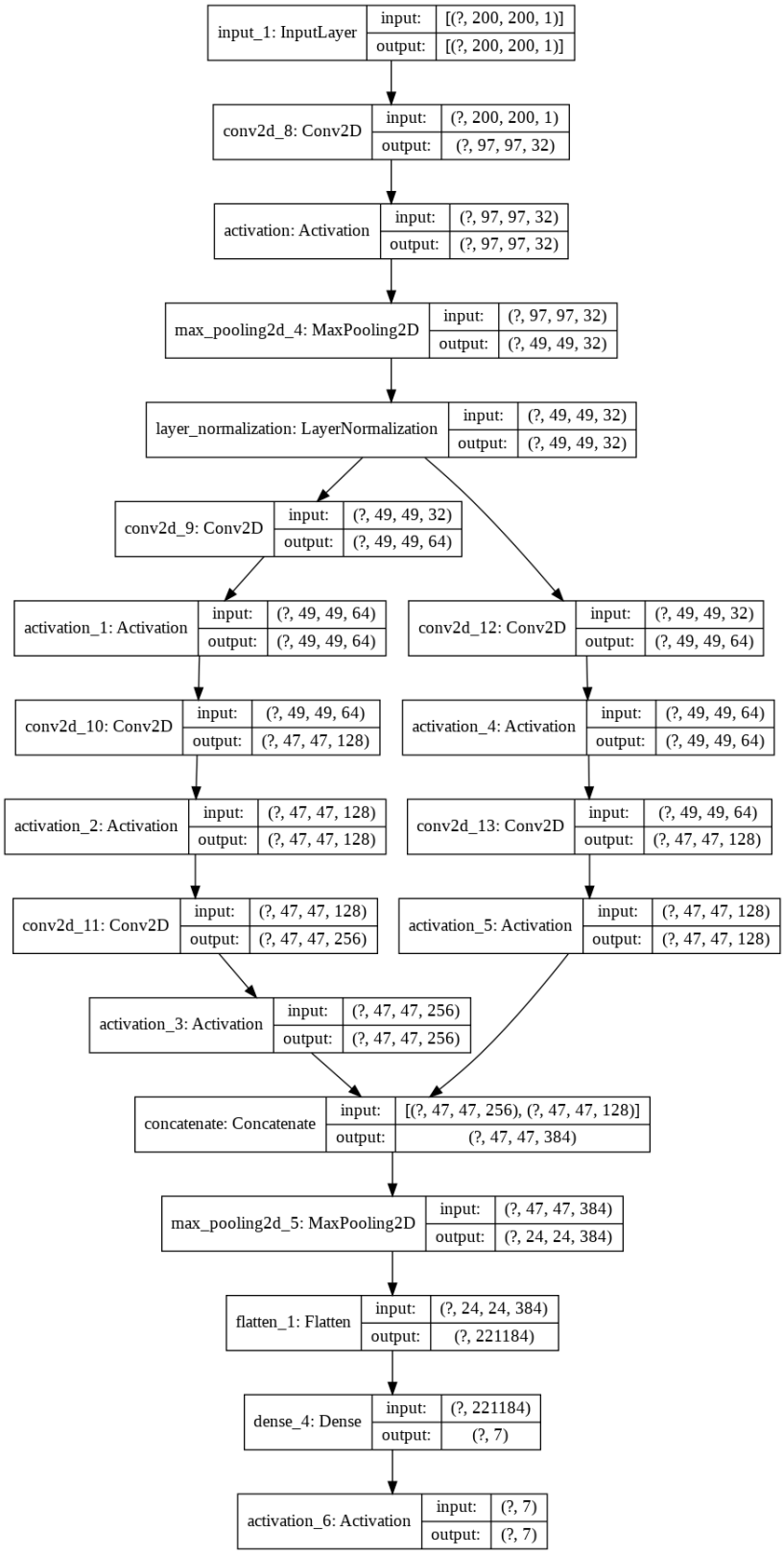

Figure 11.2: Model 2 Architecture

Modified Simple models 1, 2

The simple model that we described above had parameters over 1M, this excess of parameters was reduced in this model. In this model the total number of trainable parameters are 341,287. This is significantly lower than the previous model - almost one sixth of the simple model - and almost half of the DeXpression model. Despite the lower number of parameters, the performance of the model was on par with the DeXpression model.

We made a few changes in the number of filters in each convolutional layer to reduce the number of parameters. The 7 x 7 filters in the first layer were replaced with 5 x 5 filters. Moreover, changes were made in the parallel path too. In the first path, the number of convolutional layers were reduced from 3 to 2; removal of 1 x 1 convolutional layer. In the second path instead of 2 layers there are 3 convolutional layers: 2 layers of filters 3 x 3 and one 1 x 1 convolutional layer.

Similar to DeXpression and simple model 1, we kept the layer normalization before splitting for ensuring that exactly same number of computations are performed at training and testing size.

In another version of this model, we used batch normalization instead of layer normalization. With batch normalization, we observed that the model's performance was slighly better when tested on the testing set.

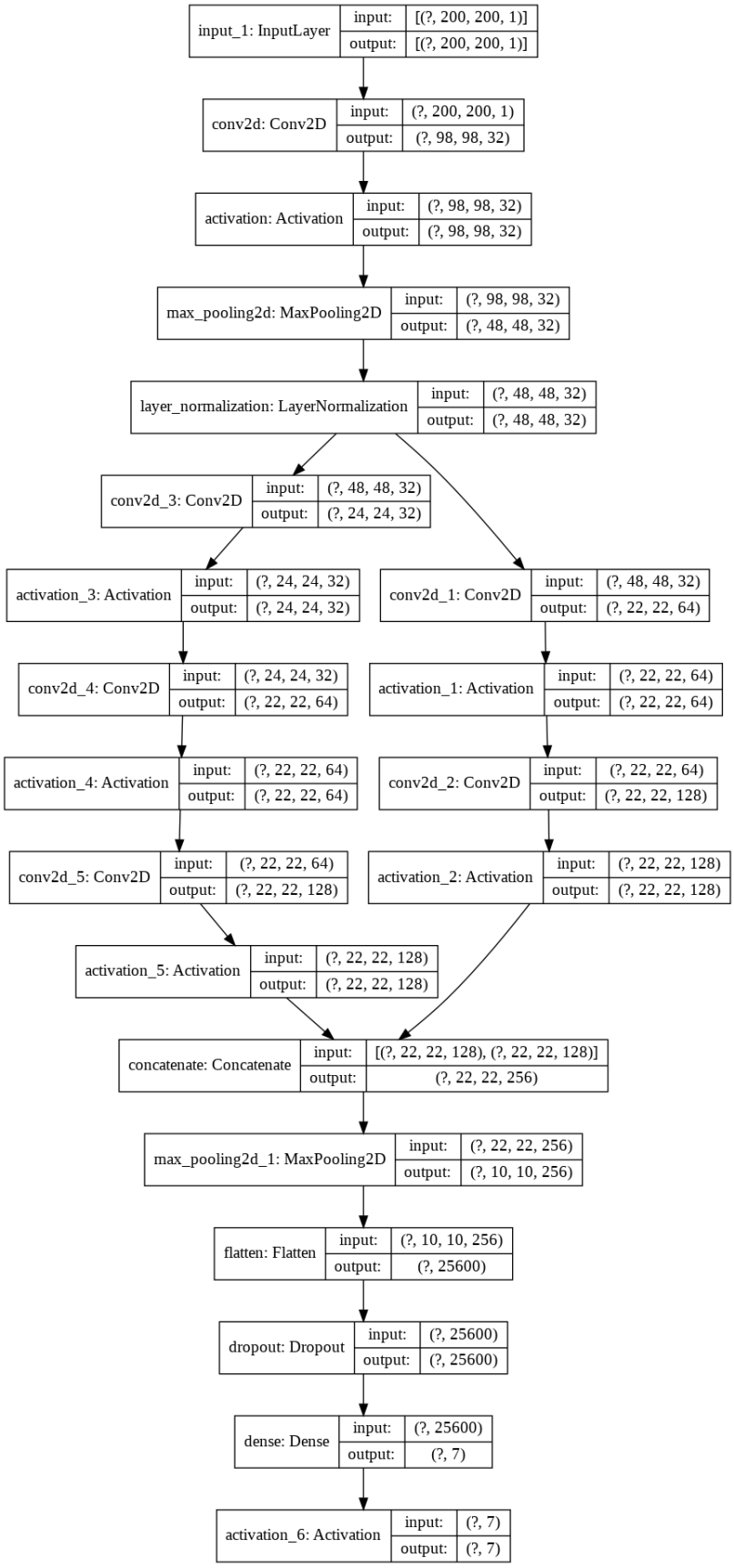

Figure 11.3: Modified simple model with Layer Normalization

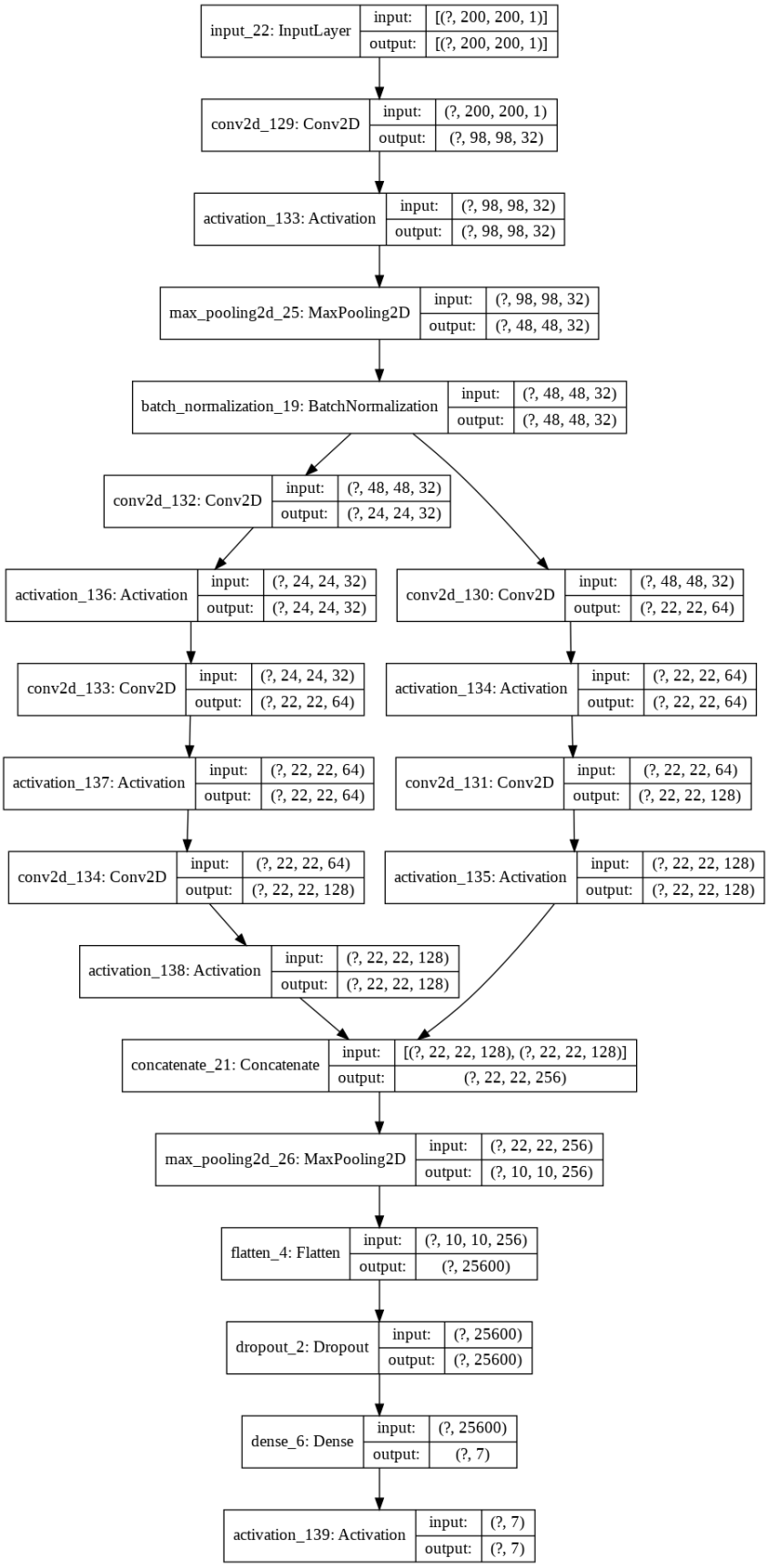

Figure 11.4: Modified simple model with Batch Normalization

11.3.2 Models trained on FER2013 dataset

The FER2013 dataset is probably the most well-known Facial Expression Recognition dataset. It contains all grayscale images in a 48 x 48 size. We used a total of 28,709 images to train our models, and 3,589 images to validate the models while training. We also used 3,589 images to test the performance of our models.

In this project, for the FER 2013 dataset we used the data augmentation to increase the data size and thus reduce the overfitting.

To use the dataset, we had to first save the csv file to a dataframe. Then using the labels provided in the file, we extracted the image data into a numpy array.

Model 1

The first model we constructed was inspired by the VGG16 architecture. The filters used in the convolutional layers are all of size 3 x 3, with 1 stride each. We have used the max pooling layer, with pool size of 2 x 2, and 2 strides, for dimensionality reduction. In each of the convolutional layer Rectified Linear Unit is used as the activation function.

The model is arranged symmetrically such that after every other convolutional layer, a max pooling layer and a dropout layer with a probability of 50% is placed. For normalizing the input of intermediate layers, we have added Batch Normalization layers.

The model is constructed using blocks of convolutional layer, maxpooling layer, batch normalization layer and dropout layer. The order of the block is as follows: a convolutional layer followed by batch normalizational layer, which is again followed by a convolutional layer and batch normalization layer. At the end of the block there is a max pooling layer of pool size (2,2) and strides (2,2). There are four such blocks. The sole purpose of these blocks is to extract maximum amount of features from the image.

At the end, fully connected layers are added after flattening of the output of the last block of convolutional layers. The output of the fully connected layers is given to the softmax layer for classification of the facial expression.

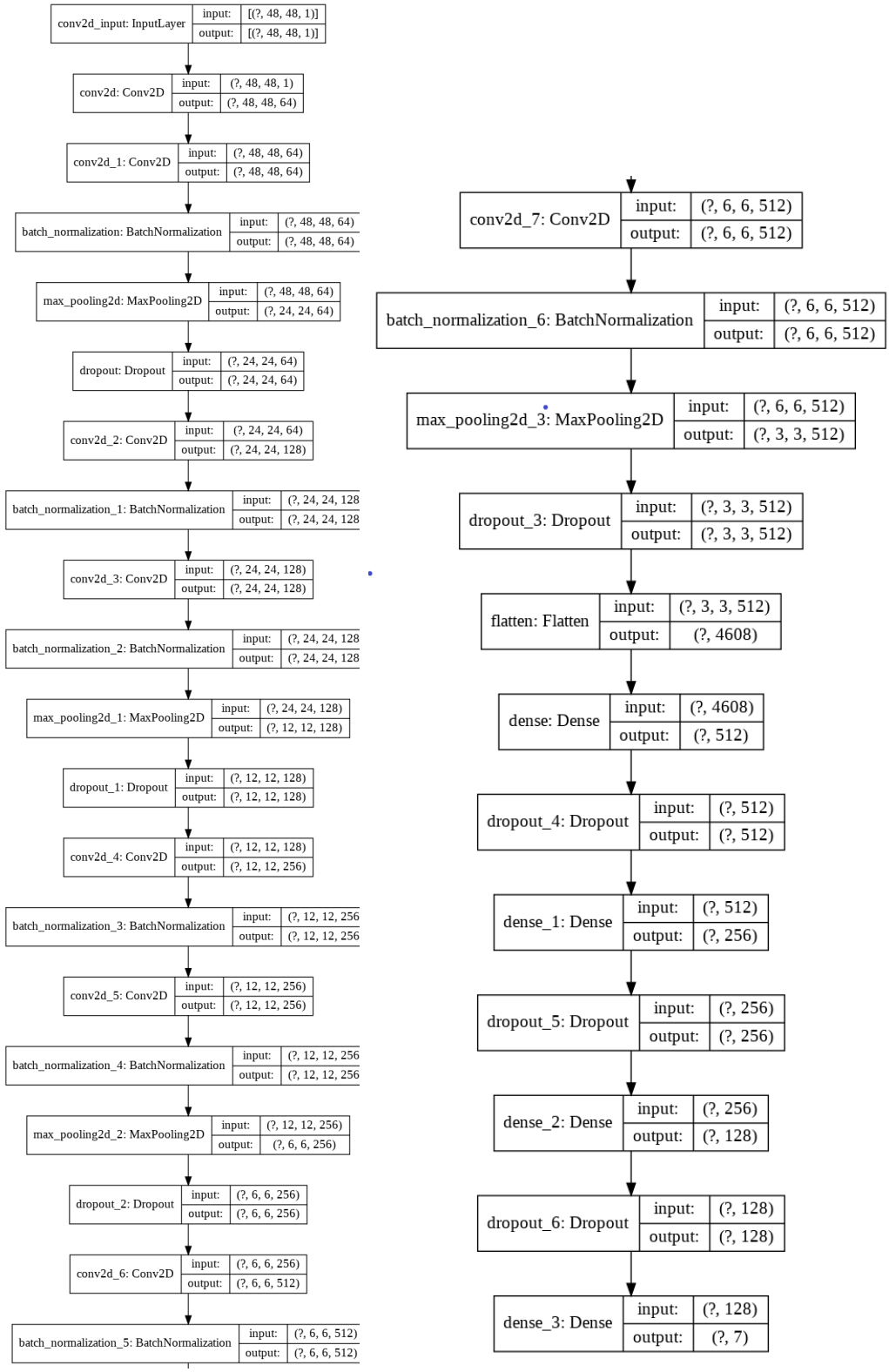

Figure 11.5: Model 1

Model 2

In the second model, we made revisions to the first. There are a lesser number of convolutional layers. Additionally, there are two dense layers, and also a reduced numner of feature maps in this model.

After every convolutional layer, there is a max pooling layer. The size of filter in every convolutional layer is 2 x 2 with a stride of (1,1). The max-pooling layer had pool size of (2,2) and stride of (2,2) in each layer. After each convolutional layer there is a batch normalization layer to normalize the outputs of the convolutional layers.

Thus, such a set of convolutional, batch normalization and maxpooling created a block. At the end of each of these blocks there is a dropout layer with a dropout probability of 0.25. There are five such blocks in the model. The number of filters of convolutional layer in each block is different. The number of filters in the convolutional layer of the first block has 32 filters and that in the layer block has 512 filters.

At the end of fifth block the output of that block is flattened and given to a fully connected layer. There are two such concatenated fully connected layers with dropout and batch normalization in between them. The output of these layers is then given to the softmax layer for generating probabilities for each categories of facial expression.

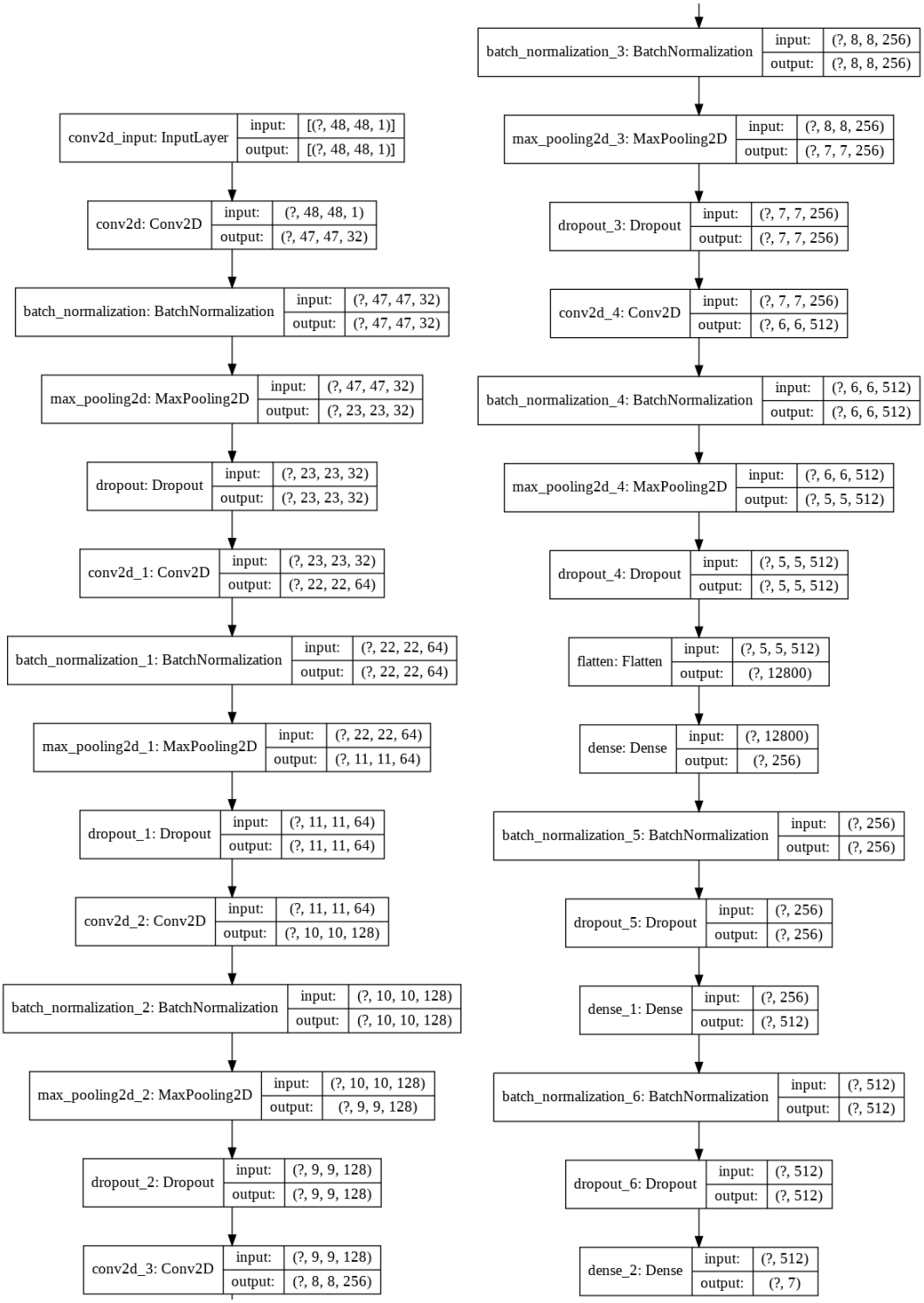

Figure 11.6: Model 2

DeXpression

The DeXpression model which is introduced in [22] is also trained in this project. It is used to compare with the other two models. This model performed effectively on the kdef dataset so it was also used in training FER 2013. We also trained the DeXpression model on the FER 2013 dataset. We added a drop out layer to reduce overfitting. In this model, the number of parameters reduced significantly in comparison to the above two models.

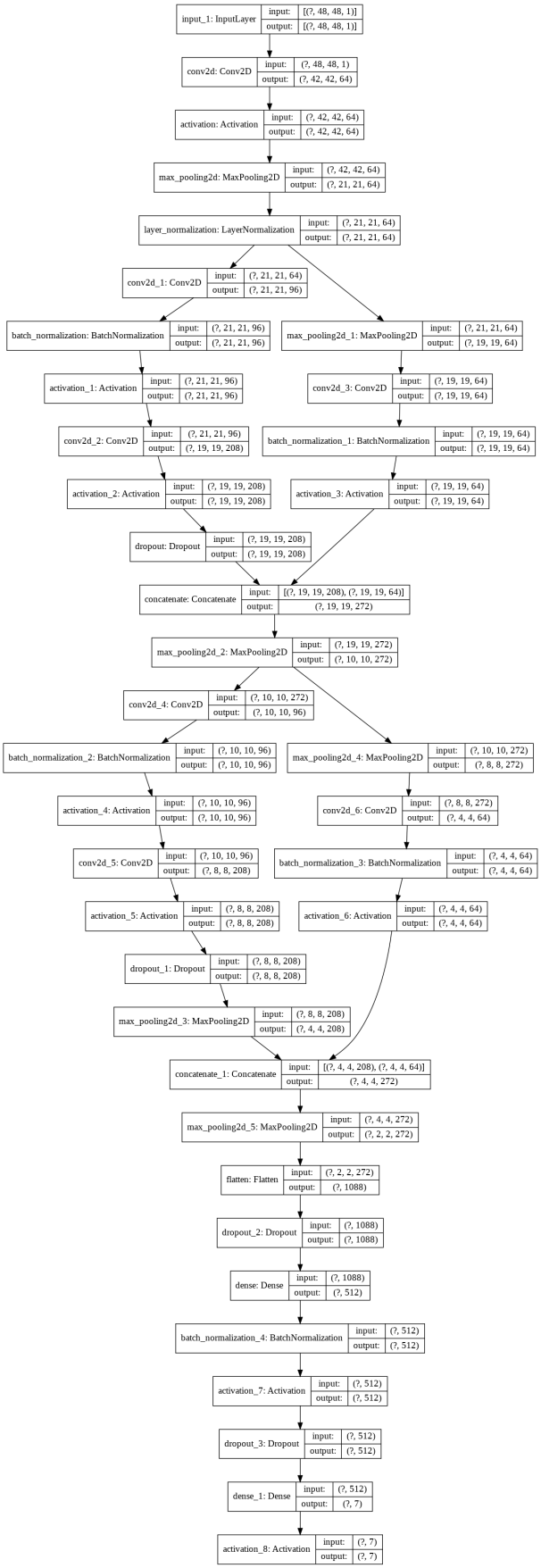

Figure 11.7: Dexpression Model on FER 2013

Simple Model with Batch Normalization

This is the same model that is used in the training of the kdef dataset, only the input size of the model is changed according to the size of the images in the FER 2013 dataset. The number of parameters in this model is significantly lesser than all other models trained using FER 2013 dataset. The results on this model were only slightly lesser than the previous models, despite the small number of parameters.

Similarly, we also trained the other version of this model (one with layer normalization). The difference between the two models was marginal, with the batch normalization gaining the edge.

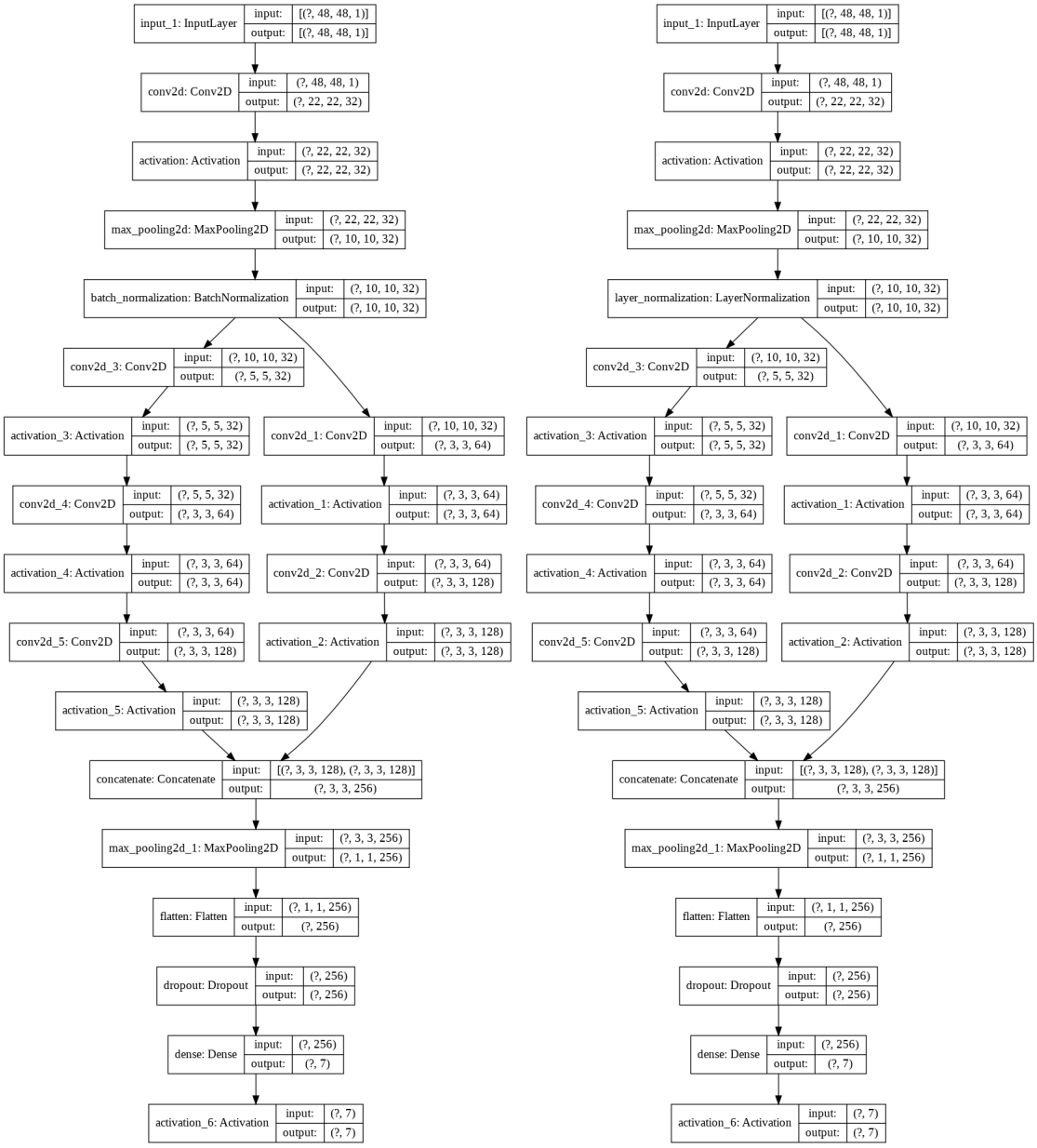

Figure 11.8: Simple Models on FER 2013

11.3.3 Model trained on JAFFE dataset

We trained one model on the JAFFE dataset. There are three sets of Convolutional layers and max pooling layers. After these, there is a combination of dropout layers and dense layers, leading to the the final fully connected layer, which predicts the expression.

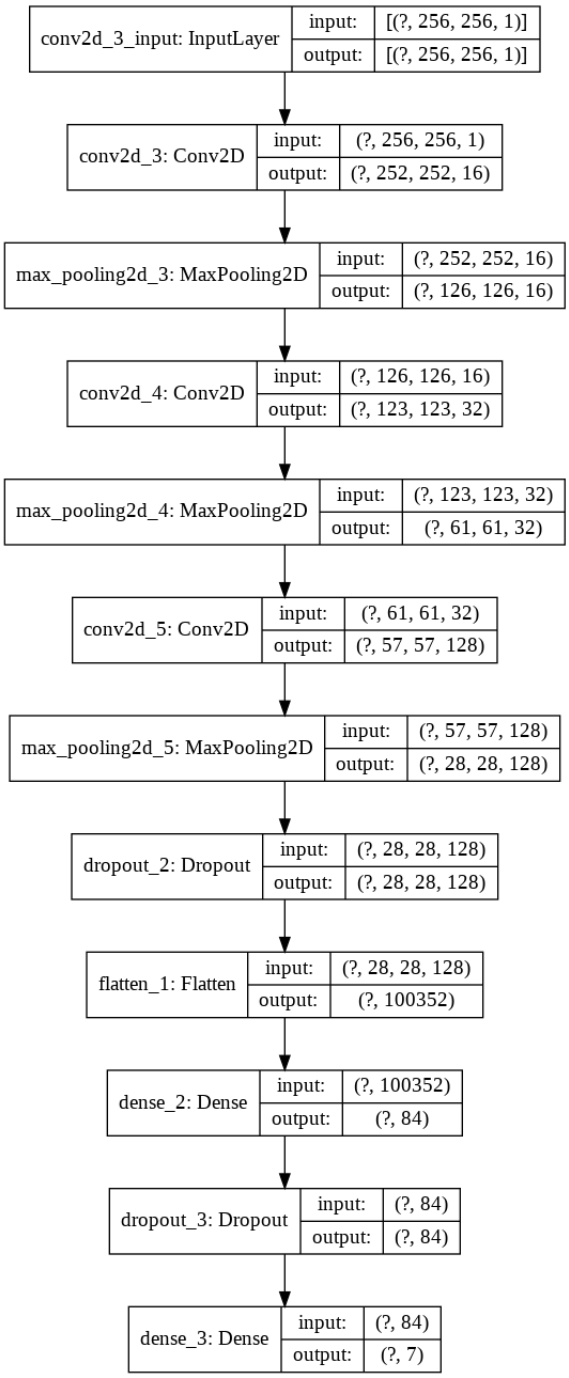

Figure 11.9: Model Structure

11.4 Callbacks

A callback is a procedure applied in various stages during training of a model on a certain dataset. These procedures can be used to view the statistics and performance of the model during training. We can define a callback and use it when we want to automate some tasks after every training/epoch that help us have controls over the training process. The callbacks that we used during training every model in this project are Early stopping, Reduce learning rate on plateau and checkpointing of a model.

- **EarlyStopping**: One way to avoid overfitting is to terminate the process early. This is a form of regularization used to reduce overfitting. Keras provides early stopping in the form of a callback. We can monitor any measure of a model using early stopping, like accuracy or loss.
- **Reduce Learning Rate on Plateau**: The reduce LR on plateau is used to reduce the learning rate whenever required during training. It can be required when there is no change in the performance of the model using current learning rate.R
- **ModelCheckpoint**: This callback is used to save the model after every epoch. There are specific measures of performance which can be monitored, and if we set the 'save_best_only' parameter as True, then the model will be saved only if there is an improvement in the aforementioned performance measure. For example, if we specify 'val_loss' as the parameter to be monitored, this callback will compare the value of val_loss with the previous epoch. If the value of val_loss has reduced, the model will be saved in the h5 file of your choosing.

11.5 Using Models for Classification on Static Images

For every model, a specific set of steps were followed to achieve the final prediction on each image. The steps involved are given below.

1. The first step is to acquire the input image on which the facial expression has to be recognized.
2. After reading the image, we have used the haarcascade frontal face detector to extract only the required part of the image i.e. the face.
3. The extracted face is converted into a grayscale image and then resized into the predefined shape of input to the model.

4. The model can now perform the classification on this processed image.
5. The output, which is the prediction for this specific image, can now be displayed to the user.

11.6 Real time and Video Implementation

The models that we trained on the static images were evaluated on the pre-recorded videos. The procedure for incorporating these models for recognizing facial expressions in videos is given below.

1. For the implementation, the python library OpenCV was used. We utilized the VideoCapture function for reading the video as input.
2. Then we have to extract each frame from the video.
3. From each frame, we have to detect the face from the whole image. The haar-cascade frontal face detector helped us in the face detection.
4. Now, this face has to be resized to match the input size of the model.
5. After preparing the face image, the model can predict the facial expression.
6. To display the outputs to the user, we have to put it directly on the video. So, as the video plays, the user can see the facial expression of each frame.
7. When we detect the face from each frame, we get the coordinates of a rectangular border around the face. Using these coordinates, we will draw a rectangle around the face on each frame in the video.
8. Finally, we will display the prediction on top of the rectangle so the user can clearly see the predicted expression of the face at each instance.

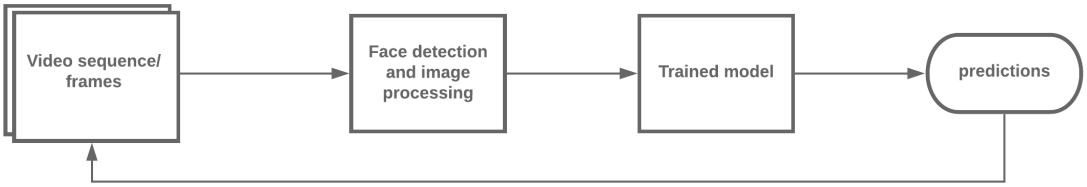

Figure 11.10: Video Processing and Prediction

11.6.1 Video implementation screenshots

This section contains the screenshots of the live video. The images show a detected face of a person and the recognized expression. The video was captured using the webcam.

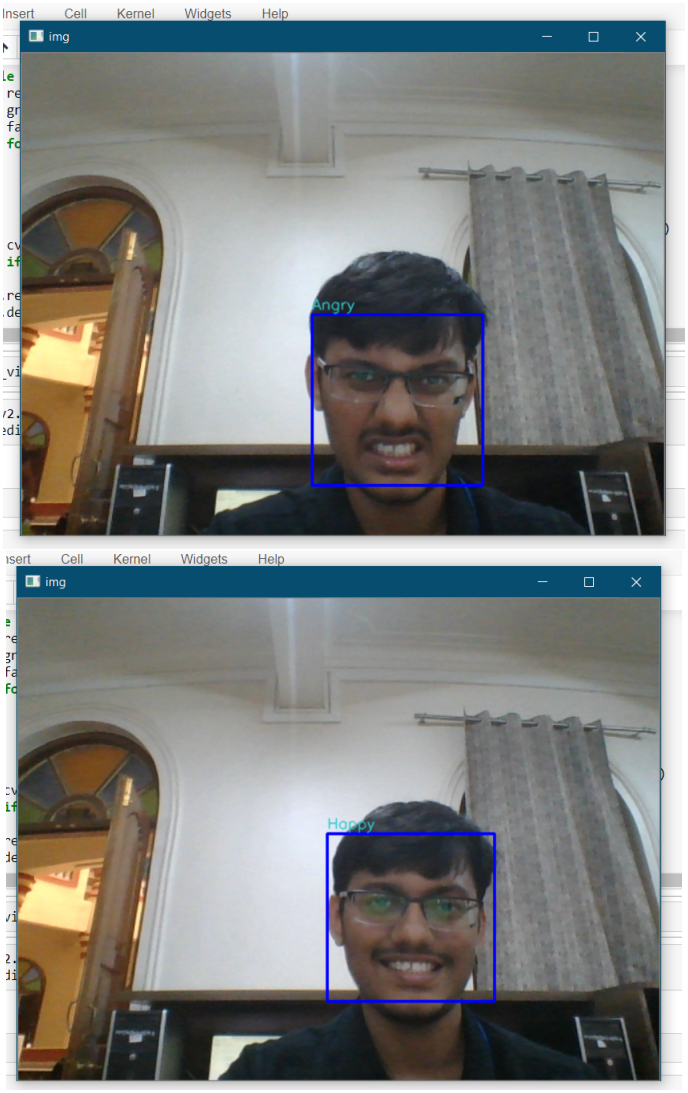

Figure 11.11: video 1

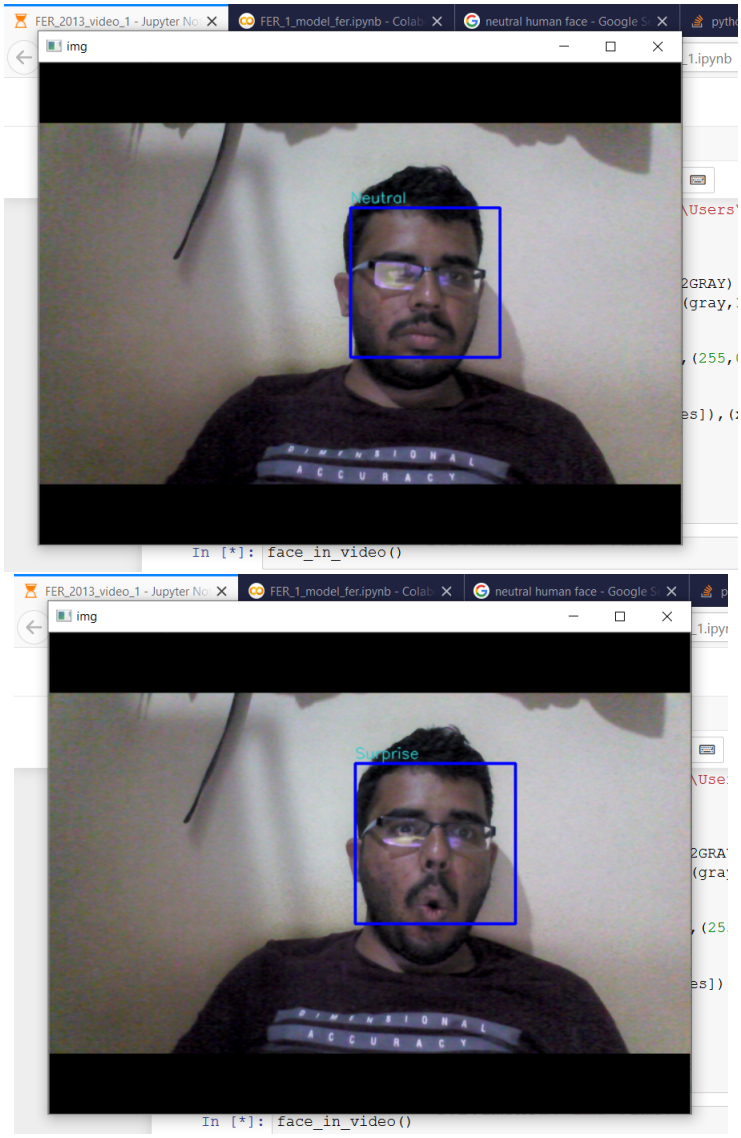

Figure 11.12: video 2

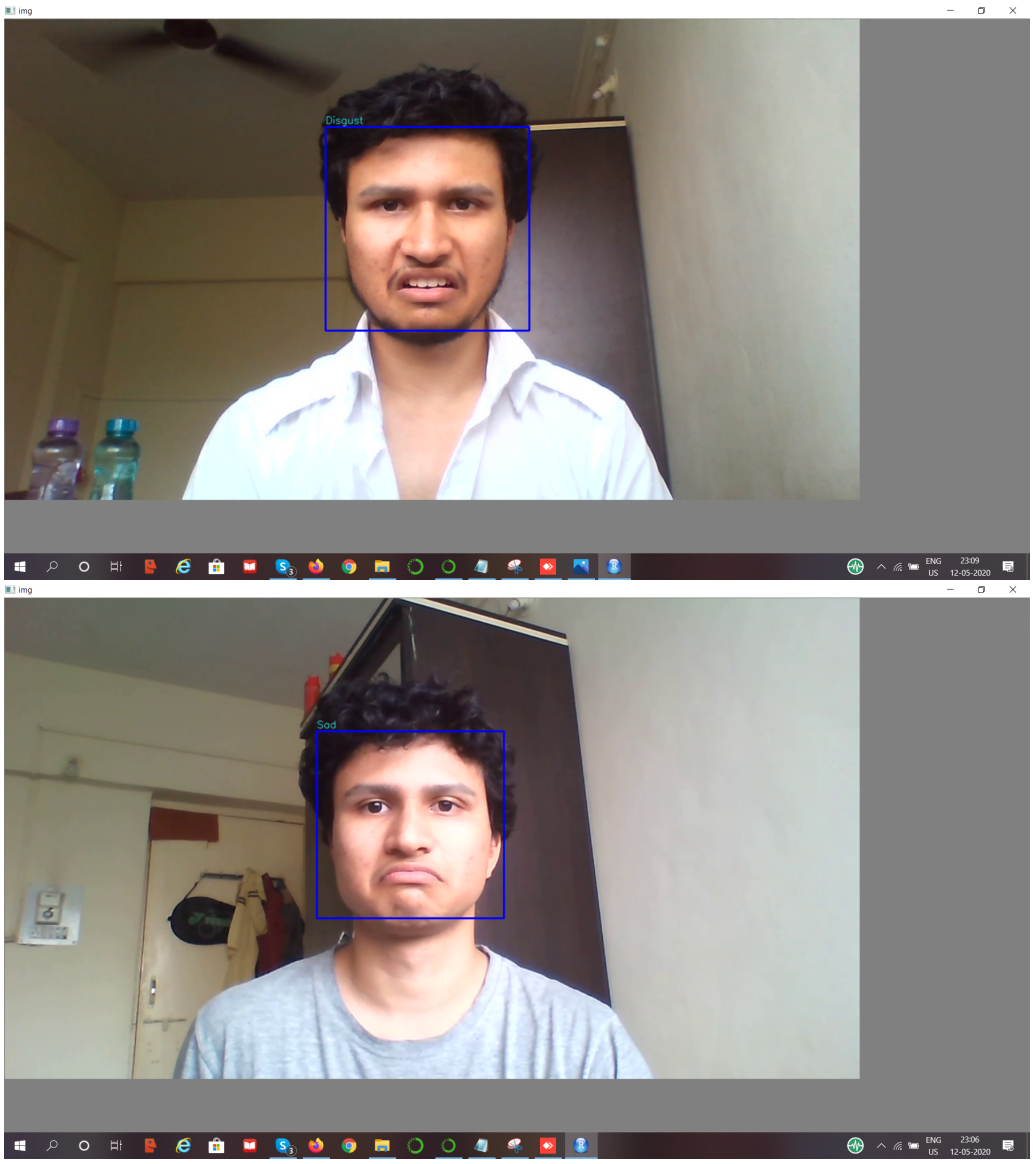

Figure 11.13: video 3

11.7 Model Deployment and Interface

In this project, we have also deployed the models that have been trained on various datasets. This deployment utilizes the FLASK API in python. With the help of FLASK, a client-server system based on HTTP request-response model is built.

When the server is active, any user access the interface of the system on his browser. On the web page he can upload the image of a person with the help of the browse button. Once uploaded, he can view the image on the section provided on the web page. Once the user sees the image of the person, he can then click the predict button to get the expression on the face of the person in the image.

When user click the predict button, an HTTP request is made to the active flask server in the backend. The request consists of the image of the person provided by the user in json format. The flask server, when receives the image, converts it into original form and sends it to the deep learning model to predict the expression. The deep learning model is attached to the server in the backend. The predicted expression is received by the server from the model. The expression is then converted into json format by the server and is sent as a HTTP response to the user. The expression is then displayed on the web page.

Below is the system architecture of the deployment. It depicts the communication between the user and the server.

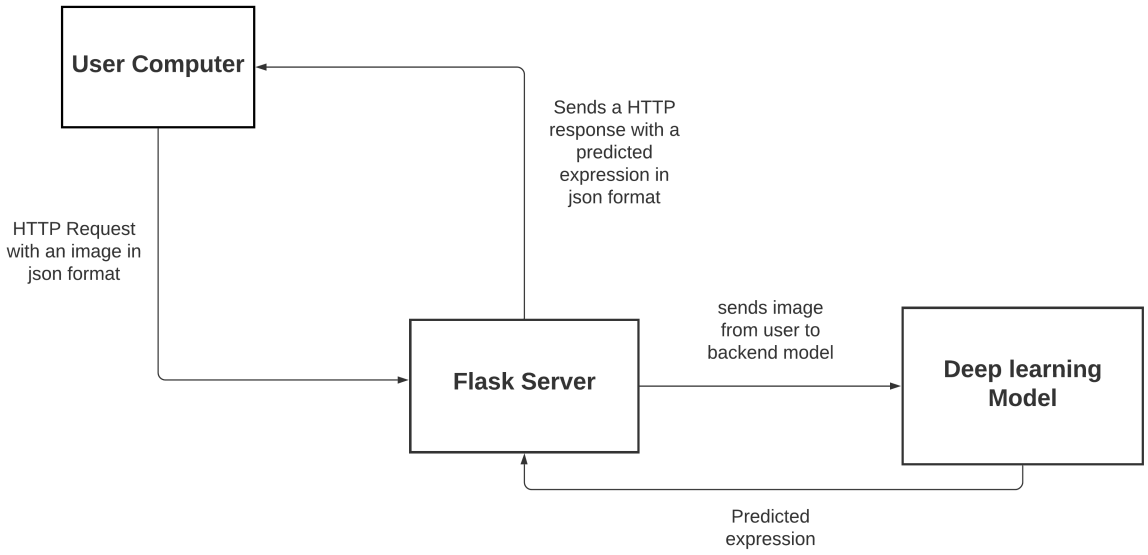

Figure 11.14: Deployment Architecture

Chapter 12

Test cases

This chapter includes the test cases that were conducted for deep facial expression recognition on our system. Test cases which are present below are the screenshots of the web page that we have created using the flask API. The test cases include images with people expressing all the seven basic expressions. The screenshots are of the web page running on model trained on the FER 2013 dataset. More specifically this is the model 1. People are of different age groups, genders and race. Not all of the predictions were correct, we have inculded some wrongly predicted expressions as well. Most of the wrong predictions are on the images with disgust, this can be attributed to the lower number of images with disgust in the training set.

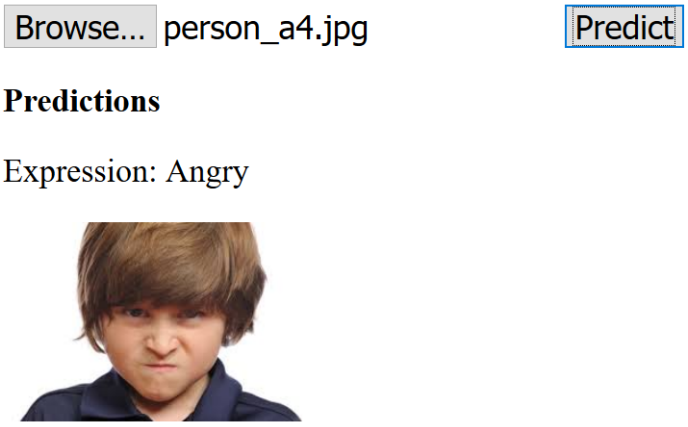

Figure 12.1: Angry predicted angry

Figure 12.2: Angry predicted angry for a man with glasses

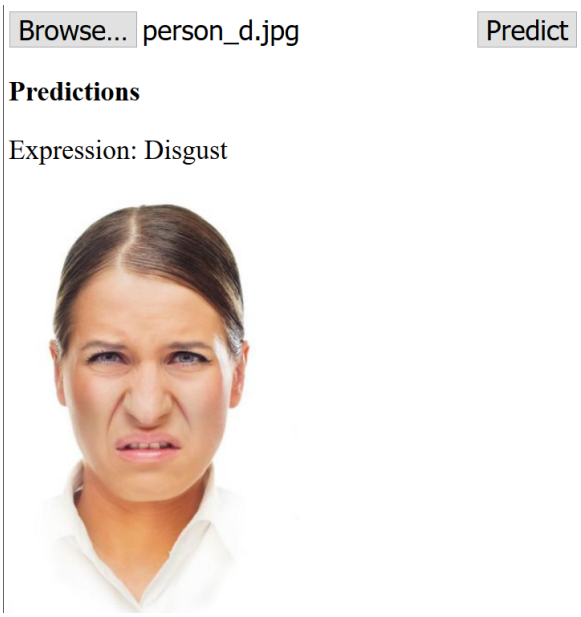

Figure 12.3: Disgust predicted disgust

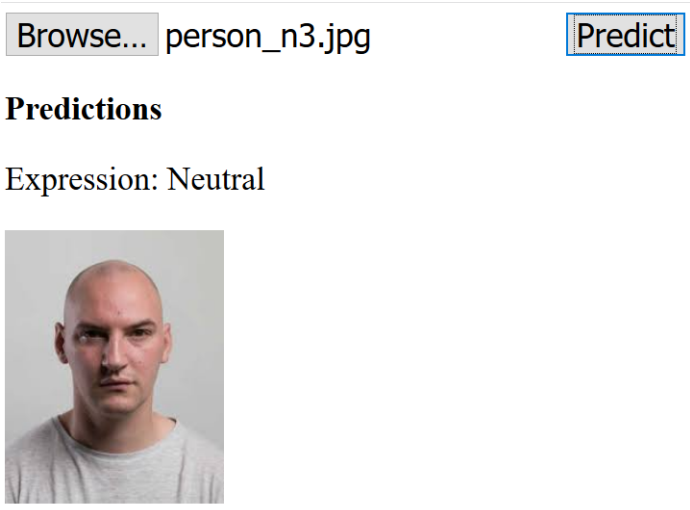

Figure 12.4: Neutral predicted neutral

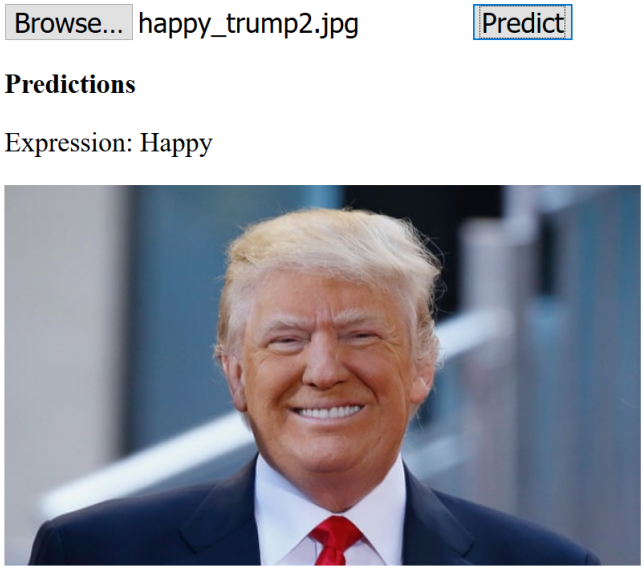

Figure 12.5: Happy predicted happy

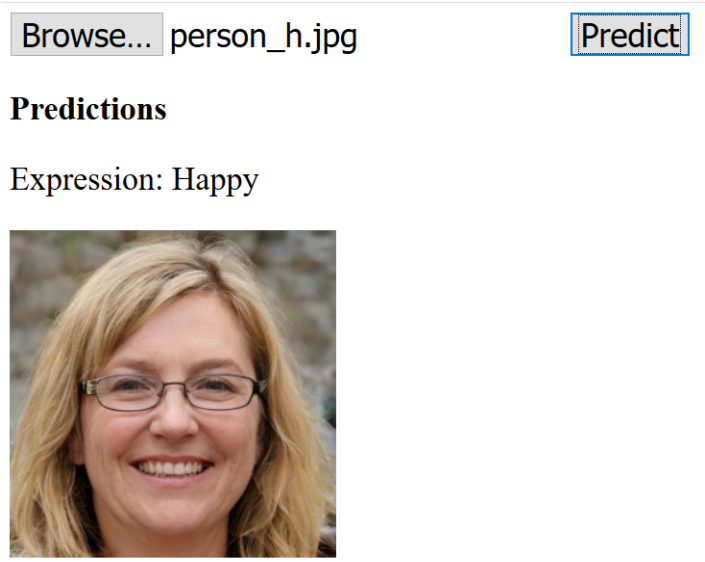

Figure 12.6: Happy predicted for a happy woman with glasses

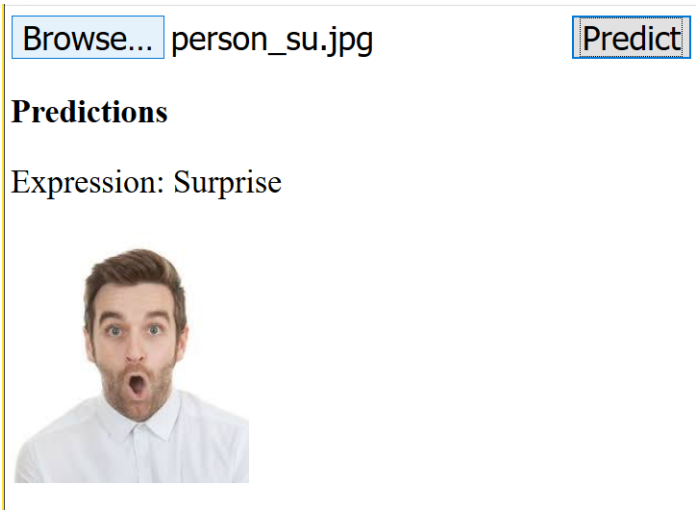

Figure 12.7: Surprise predicted Surprise

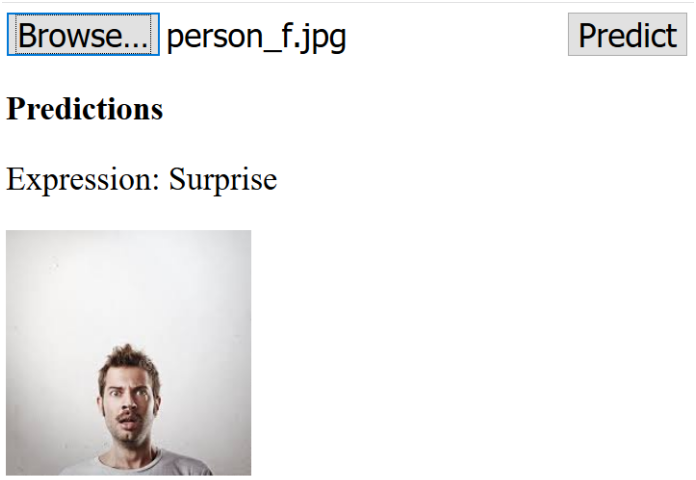

Figure 12.8: Fear predicted Surprise

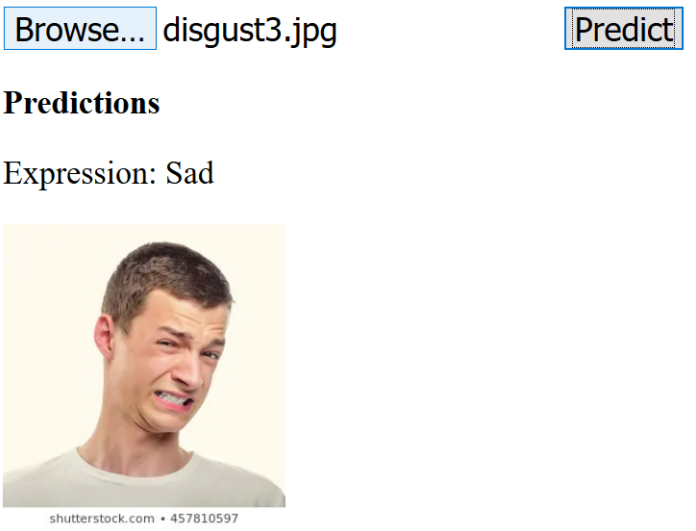

Figure 12.9: Disgust predicted Sad

Chapter 13

Experiments and Results

This chapter consists of all the results that we achieved on the models that we implemented as well as the DeXpression model. The results are orgarnised according the datasets. Each result contains the training accuracy, training loss, validation accuracy and validation loss. Along with these the graphs of epochs Vs accuracies and epochs Vs loss is also included. Additionally, the test accuracies and losses are provided for comparison.

13.1 Results on KDEF

The dataset contains 949 images of the basic facial expressions: Anger, Disgust, Fear, Happiness, Sadness, Surprise and Neutral. We have split the dataset into training, validation and testing set. The split results in 660 images for training set, 140 for validation and 149 for testing.

13.1.1 DeXpression Results[22]

Number of Model Parameters	647,639
Training Accuracy	96.06%
Training Loss	0.12
Validation Accuracy	88.57%
Validation Loss	0.4496
Testing Accuracy	85.91%
Testing Loss	0.4

Table 13.1: Results on DeXpression

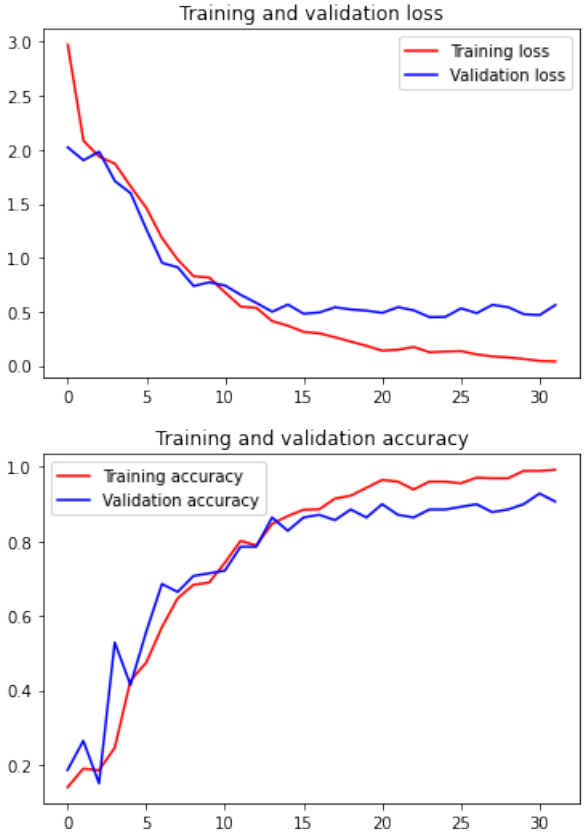

Figure 13.1: DeXpression results

13.1.2 Simple Model 1 Results

Number of Model Parameters	1,833,223
Training Accuracy	97.42%
Training Loss	0.0997
Validation Accuracy	84.29%
Validation Loss	0.5
Testing Accuracy	83.89%
Testing Loss	0.6

Table 13.2: Results on Simple model 1

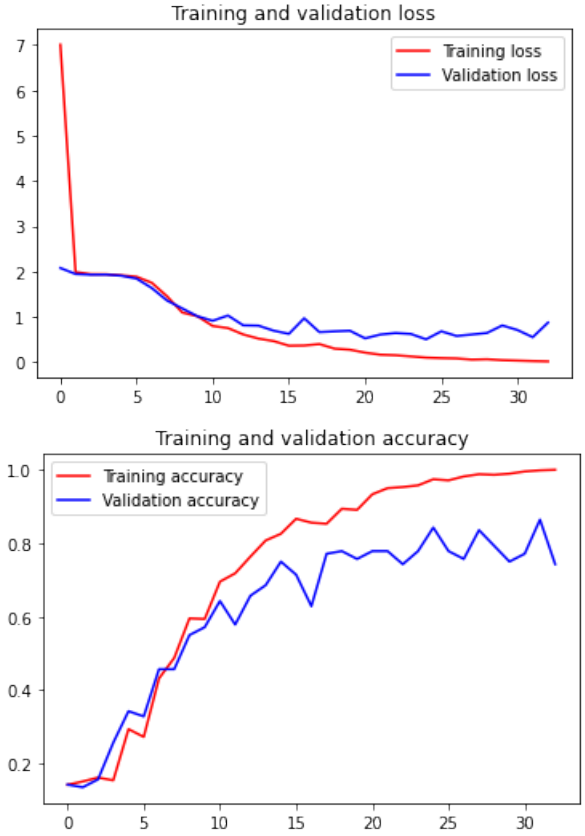

Figure 13.2: Simple Model 1 Results

13.1.3 Simple Model 2: Layer Normalization Results

Number of Model Parameters	341,287
Training Accuracy	85.61%
Training Loss	0.39
Validation Accuracy	82.86%
Validation Loss	0.45
Testing Accuracy	83.22%
Testing Loss	0.5

Table 13.3: Results on Simple model 2: Layer Normalization

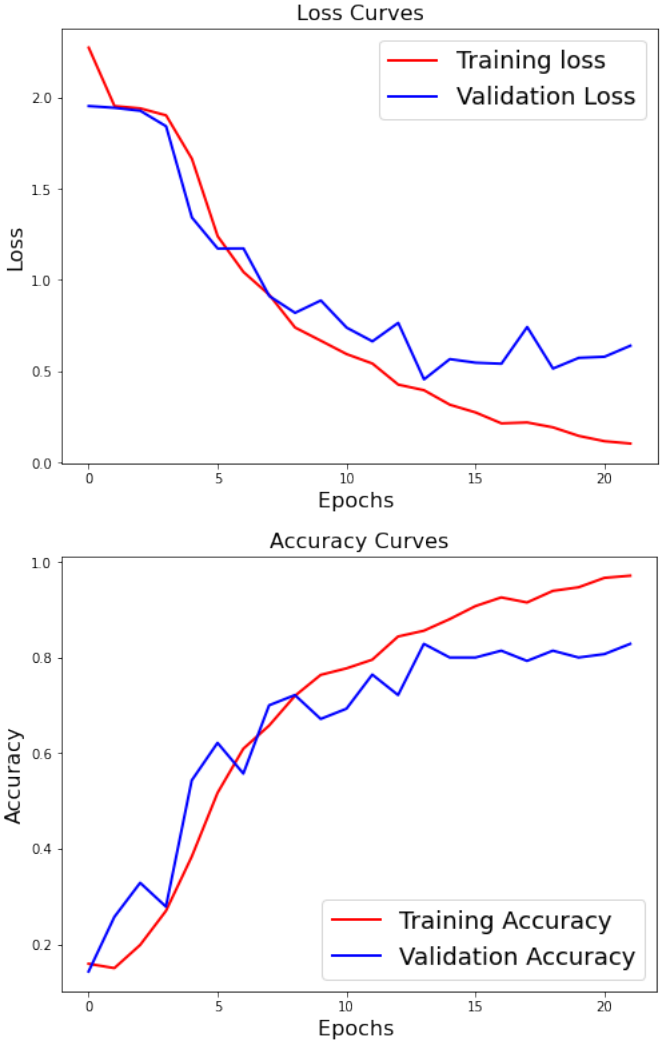

Figure 13.3: Simple Model 2: Layer Normalization Results

13.1.4 Simple Model 2: Batch Normalization Results

Number of Model Parameters	341,287
Training Accuracy	98.01%
Training Loss	0.0014
Validation Accuracy	85%
Validation Loss	0.32
Testing Accuracy	91.28%
Testing Loss	0.34

Table 13.4: Results on Simple Model 2: Batch Normalization

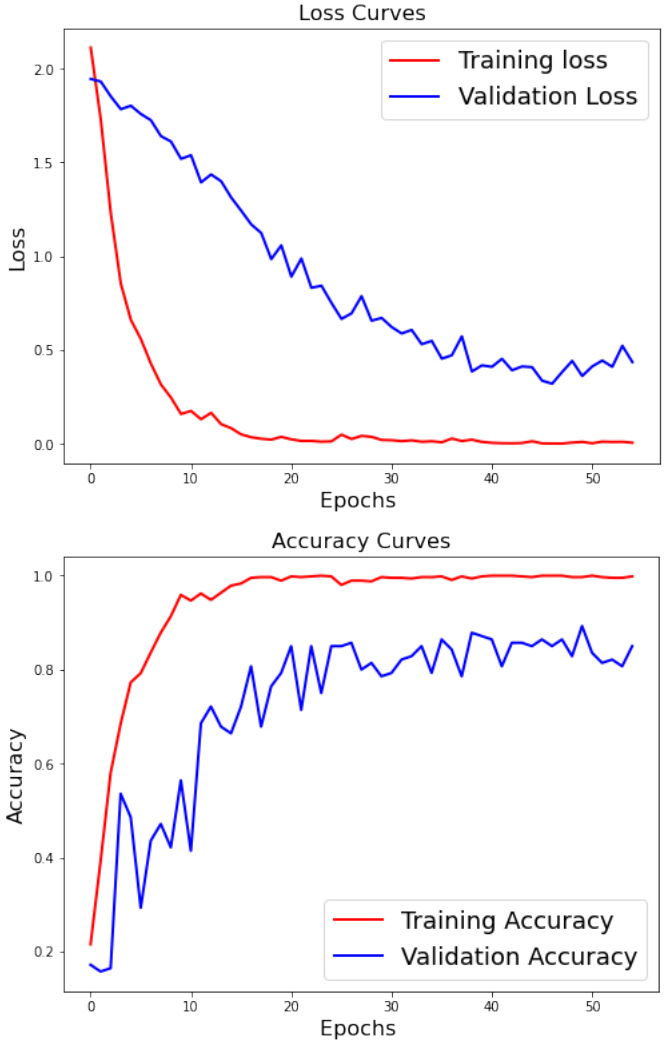

Figure 13.4: Simple Model 2: Batch Normalization Results

13.2 Results on FER 2013

The FER2013 dataset is a widely known dataset and contains 28,709 images for training set. The public test set and the private test set each consists of 3,589 examples. In this project we have used the public test set for validation and private test set for testing. Each image is grayscale and 48 x 48 x 1. The expression set of the FER 2013 dataset is: Anger, Disgust, Sadness, Happiness, Fear, Surprise and Neutral.

13.2.1 Model 1

Number of Model Parameters	7,212,871
Training Accuracy	68.12%
Training Loss	0.881
Validation Accuracy	67.21%
Validation Loss	0.8787
Testing Accuracy	64.75%
Testing Loss	0.9699

Table 13.5: Results on Model 1

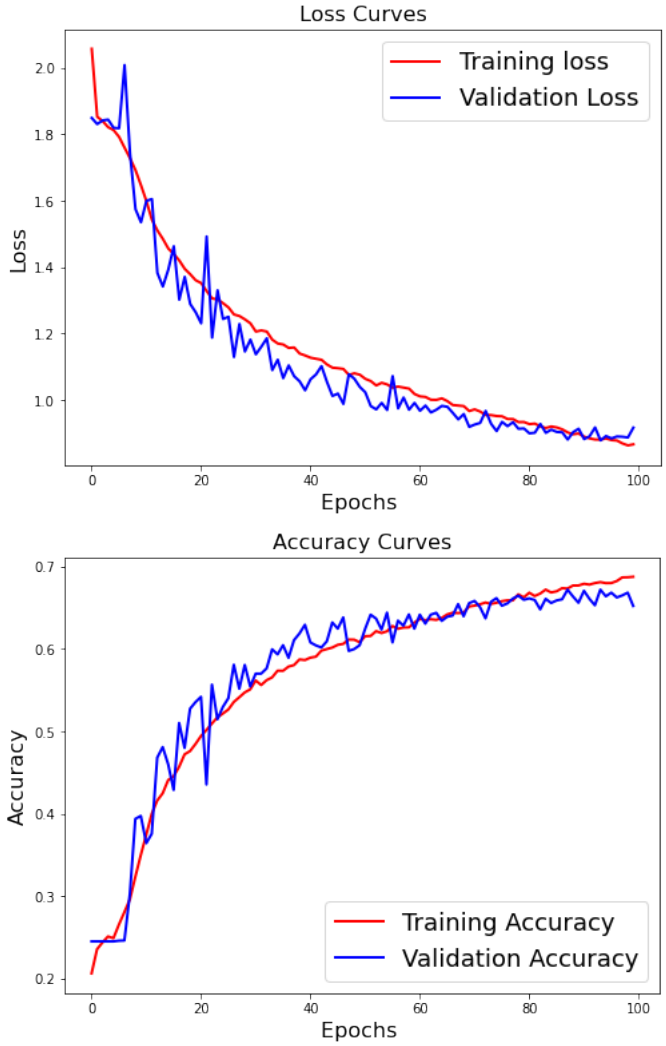

Figure 13.5: FER2013: Model 1 Results

13.2.2 Model 2

Number of Model Parameters	4,113,191
Training Accuracy	66.00%
Training Loss	0.9011
Validation Accuracy	67.68%
Validation Loss	0.8871
Testing Accuracy	66.23%
Testing Loss	0.9414

Table 13.6: Results on Model 2

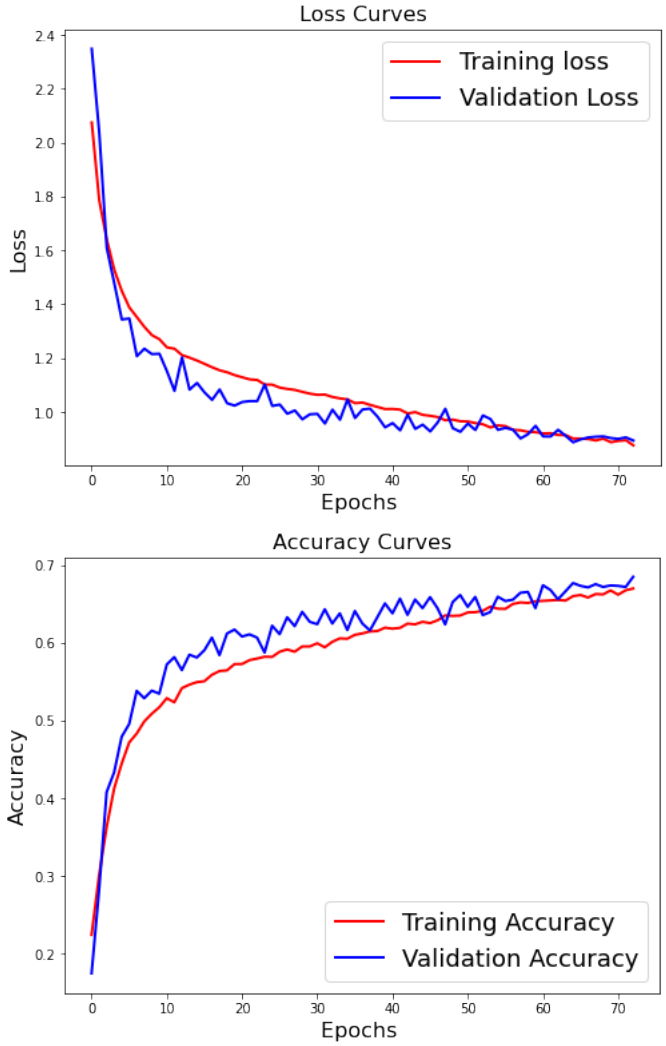

Figure 13.6: FER2013: Model 2 Results

13.2.3 DeXpression Model [22]

Number of Model Parameters	980,071
Training Accuracy	71.61%
Training Loss	0.75
Validation Accuracy	65.87%
Validation Loss	0.9177
Testing Accuracy	64.06%
Testing Loss	0.98

Table 13.7: Results on DeXpresion

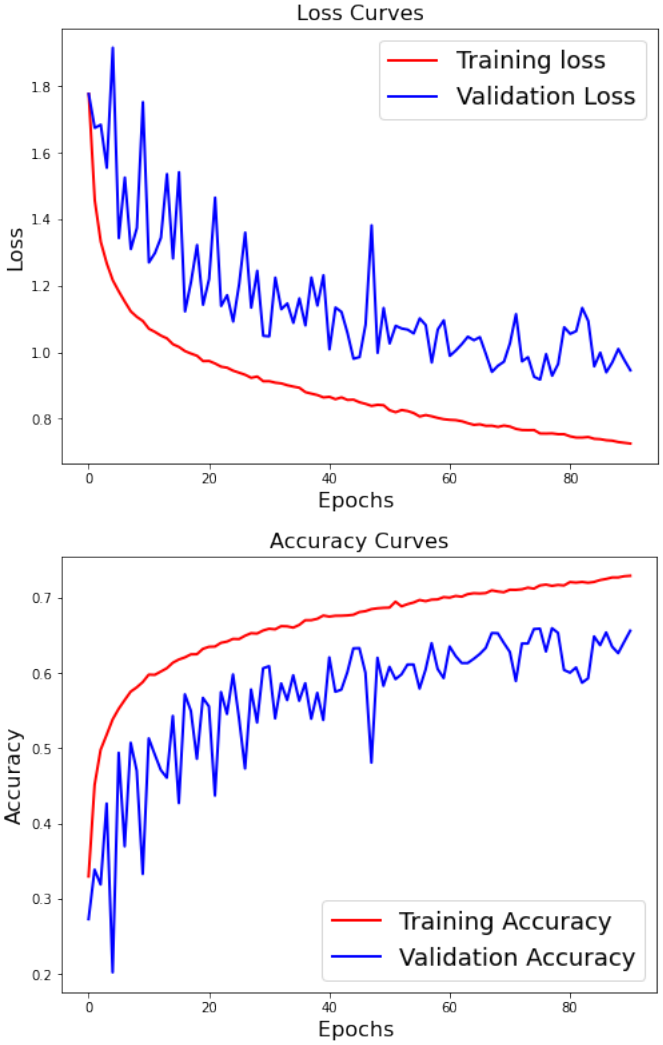

Figure 13.7: FER2013: DeXpression Results

13.2.4 Simple Model: Batch Normalization

Number of Model Parameters	163,943
Training Accuracy	56.88%
Training Loss	1.1358
Validation Accuracy	59.32%
Validation Loss	1.0784
Testing Accuracy	57.43%
Testing Loss	1.1327

Table 13.8: Results on Simple Model: Batch Normalization

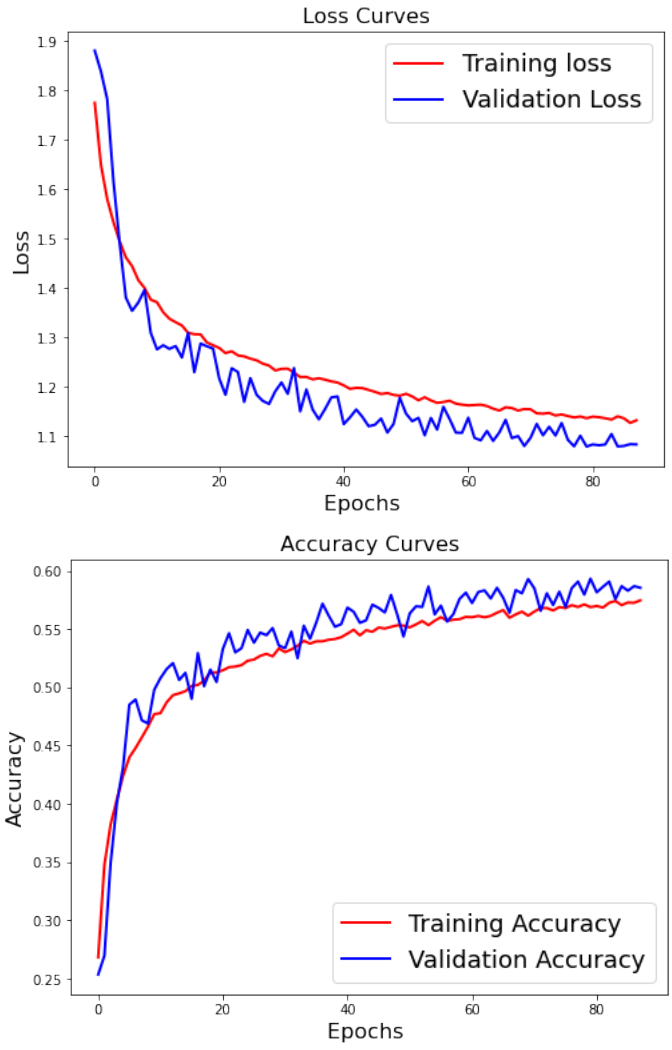

Figure 13.8: Simple Model: Batch Normalization Results

13.2.5 Simple Model: Layer Normalization

Number of Model Parameters	163,943
Training Accuracy	54.45%
Training Loss	1.2064
Validation Accuracy	56.95%
Validation Loss	1.1218
Testing Accuracy	55.92%
Testing Loss	1.1626

Table 13.9: Results on Simple Model: Layer Normalization

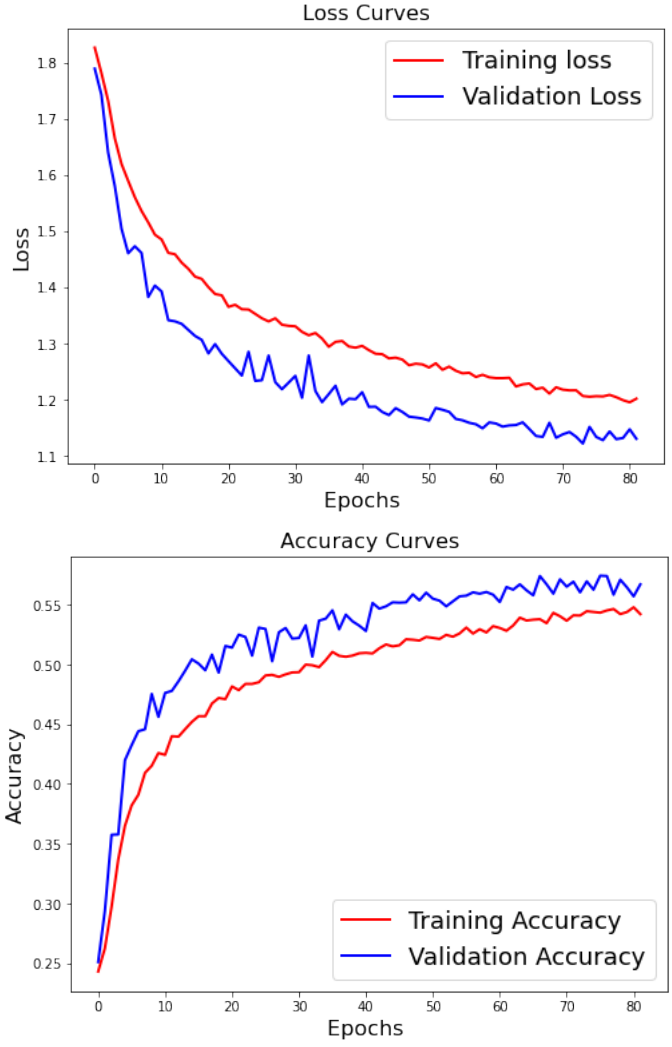

Figure 13.9: Simple Model: Layer Normalization Results

13.3 Results on JAFFE

Number of Model Parameters	8,541,415
Training Accuracy	88.41%
Training Loss	0.3026
Validation Accuracy	95.92%
Validation Loss	0.1787

Table 13.10: Results for JAFFE dataset

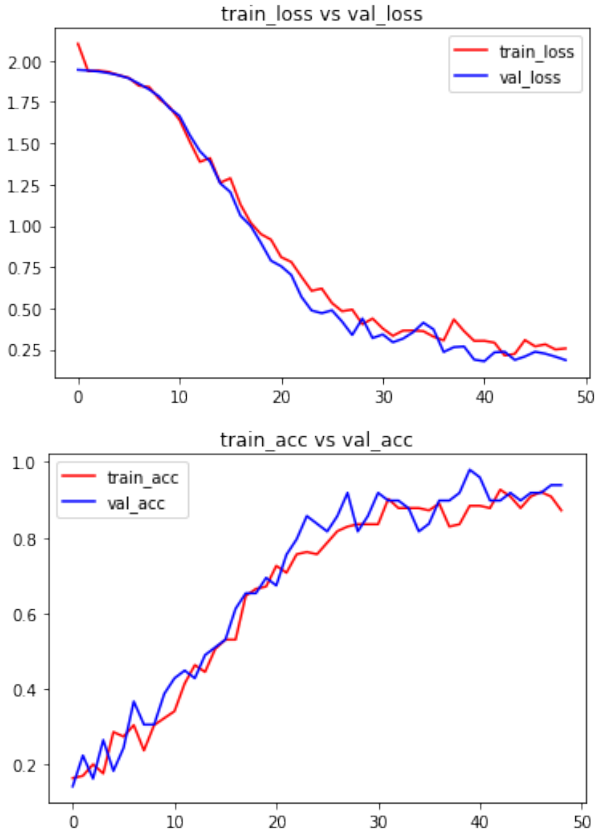

Figure 13.10: Results on JAFFE

Chapter 14

Future Scope

At present our work is mostly related to the static images and pre-recorded videos, we plan to take this forward to a real time facial expression system. We plan to implement a system that will identify facial expressions of multiple faces in the image or at real-time. We also plan on training the system on self-generated data. To accomplish this we plan to use a Generative Adversarial Network that will generate images with human faces. Further the complete system shall be deployed on an Android and iOS platform.

Chapter 15

Conclusion

After going through the recent research in facial expression recognition, it is clearly visible to us that most of the researchers are exploring deep learning methods for expression recognition. We found out that most of the models proposed, concentrate on categorising the expressions into 7 classes (6 of Ekman model and new added Neutral class). Some models described, also use Facial action units to determine the expressions on human face. It is noteworthy that the Convolutional Neural Networks are the most used architectures for feature extraction phase and are like the backbone of the entire system. Among many researches we also found out that the proposed algorithms or models tend show exceptional performance on few selected datasets like CK+, JAFFE but they underfit on difficult datasets like AFEW, SFEW and BU-3DFE.

Bibliography

[1] Pedro D. Marrero Fernandez, Fidel A. Guerrero Pena, Tsang Ing Ren, Alexandre Cunha. FERAtt: Facial Expresssion Recognition with Attention Net. Centro de Informatica, Universidade Federal de Pernambuco, Brazil. Center for Advanced Methods in Biological Image Analysis, California Institute of Technology, USA. *arXiv, 2019*

[2] Heechul Jung Sihaeng Lee Junho Yim Sunjeong Park Junmo Kim. Joint Fine-Tuning in Deep Neural Networks for Facial Expression Recognition. School of Electrical Engineering, Korea Advanced Institute of Science and Technology. *IEEE International Conference on Computer Vision, 2015*

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

[4] Debin Meng, Xiaojiang Peng, Kai Wang, Yu Qiao. Frame Attention Networks For Facial Expression Recognition In Videos. Shenzhen Institutes of Advanced Technology, Chinese Academy of Science, Shenzhen, China. Shenzhen Key Lab of Computer Vision and Pattern Recognition, Shenzhen, China. University of Chinese Academy of Sciences, Beijing, China. *IEEE International Conference on Image Processing, 2019*

[5] Oyebade K. Oyedotun, Girum Demisse, Abd El Rahman Shabayek, Djamila Aouada1, Bjorn Ottersten. Facial Expression Recognition via Joint Deep Learning of RGB-Depth Map Latent Representations. Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, L-1855 Luxembourg. Computer Science Department, Faculty of Computers and Informatics, Suez Canal University, Egypt. *IEEE International Conference on Computer Vision Workshops, 2017*

[6] Panagiotis Giannopoulos, Isidoros Perikos and Ioannis Hatzilygeroudis. Deep Learning Approaches for Facial Emotion Recognition: A Case Study on FER-

2013. *Springer International Publishing AG 2018, Advances in Hybridization of Intelligent Methods, Smart Innovation, Systems and Technologies 85, 2018*

[7] Mahesh Jangid, Pranjul Paharia and Sumit Srivastava. Video-Based Facial Expression Recognition Using a Deep Learning Approach. Manipal University Jaipur, Jaipur, Rajasthan 303007, India. textitSpringer Nature Singapore Pte Ltd. S. K. Bhatia et al. (eds.), Advances in Computer Communication and Computational Sciences, Advances in Intelligent Systems and Computing 924, 2019

[8] Xuanyi Dong, Yan Yan, Wanli Ouyang, Yi Yang. Style Aggregated Network for Facial Landmark Detection. University of Technology Sydney, The University of Sydney. *IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018*

[9] C. Fabian Benitez-Quiroz Yan Wang. Recognition of Action Units in the Wild with Deep Nets and a New Global-Local Loss. Dept. Electrical and Computer Engineering, The Ohio State University. *IEEE International Conference on Computer Vision (ICCV), 2017*

[10] Shan Li and Weihong Deng. Deep Facial Expression Recognition: A Survey. Pattern Recognition and Intelligent System Laboratory, School of Information and Communication Engineering, Beijing. University of Posts and Telecommunications, Beijing, 100876, China. *arXiv, 2018*

[11] Shiv Ram Dubey and Snehasis Mukherjee. A Multi-Face Challenging Dataset for Robust Face Recognition. *IEEE 15th International Conference on Control, Automation, Robotics and Vision (ICARCV) Singapore, November 18-21, 2018*

[12] Divya Mangala B.S and Prajwala N.B. Facial Expression Recognition by Calculating Euclidian Distance for Eigen Faces using PCA. *IEEE International Conference on Communication and Signal Processing, April 3-5, 2018.*

[13] Ramon Zatarain-Cabada, Maria Lucia Barron-Estrada, Francisco Gonzalez-Hernandez, Hector Rodriguez-Rangel. Building a face expression recognizer and a face expression database for an intelligent tutoring system. Posgrado en Ciencias de la Computación, Instituto Tecnológico de Culiacán, Culiacán, Sinaloa, México. *IEEE 17th International Conference on Advanced Learning Technologies, 2017*

[14] Jia Xiang, Gengming Zhu. Joint Face detection and Facial Expression Recognition with MTCNN. College of Computer Science and Engineering, HNUST, Xiangtan, China. *IEEE 4th International Conference on Information Science and Control Engineering, 2017*

[15] Kirti Dang, Shanu Sharma. Review and Comparison of Face Detection Algorithms. CSE Department, ASET, Amity University, Noida, India. *IEEE 7th International Conference on Cloud Computing, Data Science and Engineering - Confluence, 2017*

[16] Yuqian Zhou, Ding Liu, Thomas Huang. Survey of Face Detection on Low-quality Images. Beckmann Institute, University of Illinois at Urbana-Champaign, USA. *13th IEEE International Conference on Automatic Face and Gesture Recognition, 2018*

[17] Baohan Xu, Yingbin Zheng, Hao Ye, Caili Wu, Heng Wang, Gufei Sun. Video Emotion Recognition With Concept Selection. Zhongan Technology, Videt Tech, East China Normal University, Shanghai, China. *IEEE International Conference on Multimedia and Expo (ICME), 2019*

[18] Xianzhang Pan, Wenping Guo , Xiaoying Guo , Wenshu Li , Junjie Xu and Jinzhao Wu. Deep Temporal–Spatial Aggregation for Video-Based Facial Expression Recognition. Institute of Intelligent Information Processing, Taizhou University, Taizhou 318000, China. School of Software Engineering, Institute of Big Data Science and Industry, Shanxi University, Taiyuan 030006, China.College of information science and technology, Zhejiang Sci-Tech University, Hangzhou 310018, China.College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China.Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Guangxi University for Nationalities, Nanning 530006. *IEEE Access Volume 7, 2019*

[19] Behzad Hasani, and Mohammad H. Mahoor. Spatio-Temporal Facial Expression Recognition Using Convolutional Neural Networks and Conditional Random Fields. Department of Electrical and Computer Engineering, University of Denver, Denver, CO. *arXiv, 2017*

[20] Dinesh Acharya, Zhiwu Huang, Danda Pani Paudel, Luc Van Gool. Covariance Pooling for Facial Expression Recognition. Computer Vision Lab, ETH Zurich, Switzerland VISICS, KU Leuven, Belgium. *arXiv, 2018*

[21] Salah Al-Darraji1(B), Karsten Berns, and Aleksandar Rodi. Action Unit Based Facial Expression Recognition Using Deep Learning. Robotics Research Lab, Department of Computer Science, University of Kaiserslautern, Kaiserslautern, Germany Robotics Laboratory, Mihailo Pupin Institute, University of Belgrade, Belgrade, Serbia. *Springer International Publishing AG, 2017*

[22] Peter Burkert, Felix Trier, Muhammad Zeshan Afzal,Andreas Dengel and Marcus Liwicki. DeXpression: Deep Convolutional Neural Network for Expression

Recognition. German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany. University of Kaiserslautern, Gottlieb-Daimler-Str., Kaiserslautern 67663, Germany. *arXiv, 2016*

[23] Ai Sun, Yingjian Lij , Yueh-Min Huang, Qiong Li and Guangming Lu. Facial expression recognition using optimized active regions. School of Computer Science and Technology,Harbin Institute of Technology (Shenzhen),Shenzhen, China. *Springer, 2018*

[24] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio. Challenges in representation learning: A report on three machine learning contests. Neural Networks, 64:59–63, *2015. Special Issue on "Deep Learning of Representations"*

[25] Lucey, Patrick and Cohn, Jeffrey and Kanade, Takeo and Saragih, Jason and Ambadar, Zara and Matthews, Iain. (2010). The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. *IEEE CVPRW, 2010*

[26] Michael J. Lyons, Shigeru Akamatsu, Miyuki Kamachi, Jiro Gyoba. Coding Facial Expressions with Gabor Wavelets. *IEEE ICAFGR, 1998*

[27] Lundqvist, D., Flykt, A., and Öhman, A. (1998). The Karolinska Directed Emotional Faces - KDEF, CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet.

[28] Abhinav Dhall, Roland Goecke, Simon Lucey, Tom Gedeon, Collecting Large, Richly Annotated Facial-Expression Databases from Movies, IEEE Multimedia 2012

[29] Abhinav Dhall, Roland Goecke, Simon Lucey, and Tom Gedeon. Static Facial Expressions in Tough Conditions: Data, Evaluation Protocol And Benchmark, First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies BeFIT. *IEEE ICCV, 2011*

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. Going Deeper with Convolutions. *ILSVRC, 2014*

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. *ILSVRC, 2015*

[32] Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ICLR, 2015*

[33] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions.

[34] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *ILSVRC, 2012*