Emotion Recognition From Facial Expression
Project Final Report

Nur Bengisu Cam
21627097
Hacettepe University
bengisu.cam@gmail.com

Ayca Meric Celik
21627103
Hacettepe University
aycameric.celik@gmail.com

Ahmet Tark Kaya
21527156
Hacettepe University
ahmet.tarik.kaya@gmail.com

Abstract

Emotion detection has always been an important topic to investigate since the emotions are everywhere in our life. Humans use emotions from expressing their feelings to understanding and predicting the actions. But identifying the emotions can be hard sometimes. There are lots of different types of emotions and expressing them can vary from person to person. There are many research about emotion recognition to make life easier with increasing the understanding of the emotions. There are several methods to recognize them. For example, the overall scene or body shape can be used. In our study, we are going to use facial emotion detection, so we are only interested in facial information.

1. Introduction

We are all humans and we all have feelings. Feelings have an important role in our daily lives. We want to understand the feelings of the person we are currently talking and we want to predict the actions. We use the emotions to understand the mood or the mental health of the people. Sometimes understanding the feelings of the person we are talking to can be hard. The reason is each of us expressing our feelings differently. This makes communication between people complicated.

Estimating the emotions are even harder for the machines. There are other steps to be followed to be able to detect them. First of all, the face should be detected from the whole image. But machines cannot look through the whole image and understand as we do. So after detecting the face, the parts of the face such as eyes, eyebrows, nose, mouth need to be extracted. They have the main information about emotions. Considering how they look like such as their position and shape, machines can estimate the emotions.

As technology improves day by day, face detection started to have an important role in many applications. These applications can be used for security by detecting and recognizing faces. There are other applications of face and emotion detection which are used in the health area. Emotions can reveal about mental health. There are some mental diseases which can be understood by looking at the changes of the emotions. These diseases can be depression, mood disorders, and anxiety disorders. Everyone has mood changes in a day and sometimes these changes can go for a couple of days, which is normal. But if a mood stays permanent for a couple of weeks then this can be the sign to mood disorders. With the help of health detection systems, patients can be watched for a couple of weeks to understand if they have a normal mood change or they have mood disorders. Sometimes opposite can happen; high level of certain emotions can cause some other physical diseases and symptoms. Such as frequently high level of anger can cause stomach issues or sadness can cause cancer in later times.

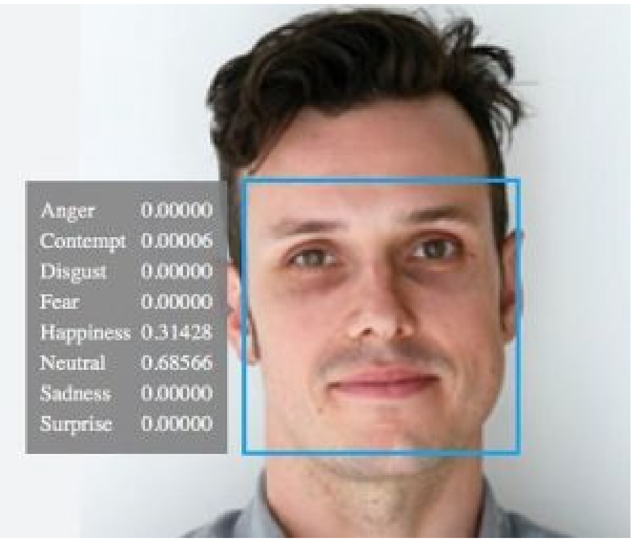

Figure 1. Calculated possibilities of every emotion for the face in the image

In this study, we will be doing emotion detection which can be used for these kinds of health systems.

2. Related Work

2.1. Emotion Recognition Approaches

To solve the automatic emotion recognition problem, various techniques from multiple areas have been used, such as signal processing, computer vision, and machine learning. It involves the analysis of human expression in multimodal forms, such as video, audio, text, image, etc. Facial expressions, speech, body movements and gestures are used to detect different emotion types.

The existing approached can be generalized into three categories:

1. **Knowledge-Based Techniques:** It is also known as lexicon-based techniques. These techniques detect certain emotion types by using the semantic and syntactic characteristics of language.

2. **Statistical Methods:** It involves deep learning and supervised machine learning algorithms. It gives more reasonable classification accuracy compared to other approaches, but having a large dataset is a must. Commonly used machine learning algorithms are Naive Bayes, Support-Vector Machines (SVM), and Maximum Entropy.

   Deep learning methods are also widely used in this task. Generally, different architectures of Artificial Neural Network(ANN) is used, such as Convolutional Neural Network (CNN), Long Short-term Memory (LSTM), and Extreme Learning Machine (ELM). These approaches have high success rates in fields of computer vision, speech recognition, and natural language processing.

3. **Hybrid Approaches:** It is a combination of knowledge-based techniques and statistical methods.

2.2. Emotion Recognition in Computer Vision

In computer vision, there have been various works which use different approaches.[4]

- In the study of Bartlett et al. [11], classification was made by using SVM with linear and RBF(Radial Basis Function) kernels. The maximum accuracy of SVM is 88%. An AdaBoost-based procedure is also used in feature selection. The approach which uses AdaBoost features and SVM classifier, named AdaSVM, gave the maximum accuracy of 90.7%. Both SVM and AdaSVM used RBF kernels and leave-subject-out approach.

- Ko [6] used deep learning methods, CNN-based (Convolutional Neural Network) and a hybrid CNN-LSTM-based (CNN-Long Short-Term Memory), for this task. The hybrid approach solves the problem of reflecting temporal variations in facial components.

- Duncan et al. [3] has another study which implemented CNN. It uses VGG_S structure, CK+ (Extended Cohn-Kanade) and JAFFE(Japanese Female Facial Expression) databases. They used Haar-Cascade filter from OpenCV to find faces on the screen. Then, they fed the VGG_S structure to obtain softmax outputs of 6 different emotions, which were angry, fear, happy, neutral, sad, and surprise. They got an average accuracy of 90%.

- Tarnowski et al. [12] carried out their experiments for 7 different emotions (neutral, joy, sadness, surprise, anger, fear, disgust). They used k-NN and MLP neural network for the classification task. They got average accuracry of 90% on MLP and 96% on 3-NN when the classification is subject dependent. The accuracy of subject-independent classification for natural division of data is 73% on MLP and 63% on 3-NN.

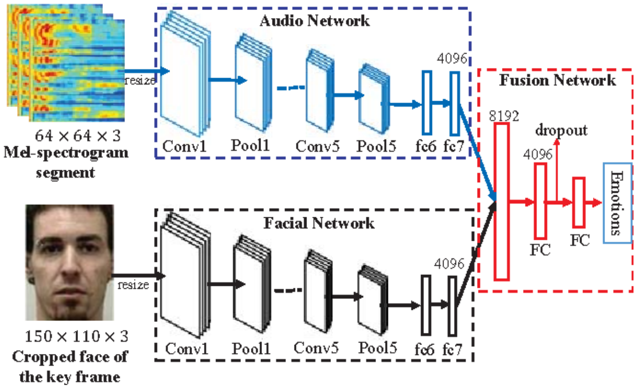

Figure 2. An example structure of a hybrid approach, which analyses both audio and image for emotion classification.

As we see, there are many promising studies for the emotion recognition task. Yet, the research continues to solve challenges such as orientation, lighting, and dataset size.

3. Method

Our goal is to recognize emotions from people's faces. To achieve that, first we should detect faces in the image. We're planning to use OpenCV's HAAR filter for that part. After detecting faces using HAAR filter and Cascade Classifier from OpenCV, we're going to cut the image and resize it. These operations are important for us because classifiers for emotion prediction will work best if the images are sized the same. Having most of the image covered by the face will lead to better results since we're going to analyze faces for prediction. At this point, there are two options for our development process: using a built-in method or create our model for predicting emotions by analyzing facial information.

In terms of implementation effort, creating a model from scratch is a complicated task. Thus, we will use a built-in method to obtain our first results. After some research, we found out that OpenCV's "Fisher Face Classifier"[2], which is an enhancement of the Eigenface method, would be the most proper implementation for the beginning. It uses linear discriminant analysis to find facial features and mainly used to distinguish people from each other. When we train this model with images of faces along with emotion labels, this classifier becomes useful for emotion recognition as well.

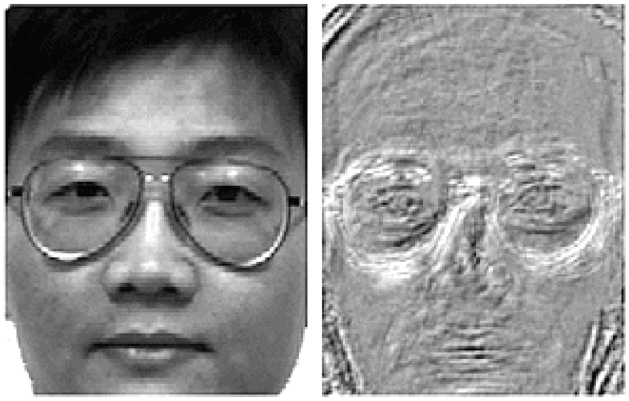

Figure 3. Original image on the left, the image that is used after operations of Fisherface method on the right

After implementing Fisherface to our model, our goal is to create our model using deep learning. Even though Fisherface can be used for emotion recognition, we foresee that it will give unsatisfying results in some situations. Fisherface recognizes face features, but it's built to use this information to distinguish people. So, analyses are not done based on how facial muscles moves based on emotions. This shortcoming might fail to classify emotion in the face so our model will classify emotion based on facial actions.

Our dataset consists of images from scenes. The first image of a scene provides the face of the subject's neutral expression, this neutral expression turns into emotional expression frame by frame. We are going to detect parts of the faces from the first image and observe actions happened on them by comparing it to the last image. This way, our model will learn which emotion results in what kind of actions in facial parts which are called action units[8]. For example, a study[9] on facial expressions found that orbital tightening, levator contraction, eye closure, and brow lowering happens when the person is in pain. These kinds of information will be observed by our algorithm and later used to detect emotion in the face.

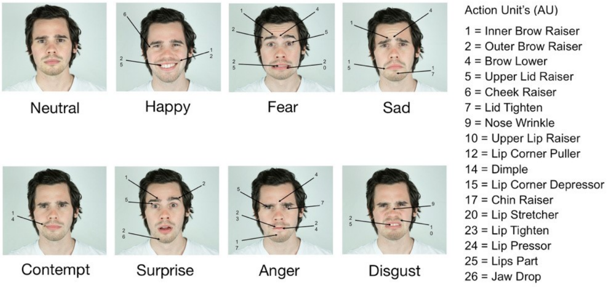

Figure 4. Action Units for happiness, fear, sadness, contempt, surprise, anger and disgust

4. Initial Experimental Settings

4.1. Our Dataset - CK+

For our project, we will use Extended Cohn-Kanade (CK+) database, which is an extended version of Cohn-Kanade (CK) database[8].

In 2000, CK database was released to create a data source for the research of automatic detection of individual facial expressions. Although this database is widely used since then, there were several issues which creates limitations. Thus, the database is revised and extended version CK+ is released in July 2010.

Nowadays, automatically detecting facial expressions becomes more and more popular research area. It can only be solved with the combination of behavioral and computer sciences. Facial expressions can be obtained by using computer vision and deep learning methods and linked with the emotions. This research area widely used in different fields, such as security[10], health-care[7], human-computer-interaction[13], and driver safety[14]. Although significant progress has been made, there are some common limitations:

1. Inconsistent or absent reporting of emotions metadata.
2. Common performance metrics for both AU and emotion detection algorithms.
3. Standard protocols for common databases.

These factors have made bench-marking different systems very difficult or impossible. These problems are highlighted in the use of the Cohn-Kanade (CK) database[5].

CK database contains 486 sequences, which is frames of a video, across 97 subjects. Each sequence contains images from onset (neutral frame) to peak expression (last frame). To obtain facial action units(AU), the peak frame was reliably FACS coded. The Facial Action Coding System (FACS) refers to a set of facial muscle movements which correspond to a displayed emotion.

CK+ database contains another 107 sequences and 26 subjects. The peak frames are FACS coded in these extended sequences as well. Emotion labels are revised, different methods are added to evaluate the performance.

In summary, CK+ database contains 593 videos and 123 subjects. Each video has 10 to 60 scenes. The last frame of each sequence, which is frames of the video, is FACS-coded to obtain posed facial expressions.

4.2. Our Dataset - FER+

FER+[1] is an enhanced version of the FER dataset. It is widely used for facial emotion recognition applications since 2013. It is a large dataset that contains 48x48 grayscale face images which are labeled with one of the following 8 emotion types: happiness, surprise, neutral, sadness, anger, disgust, fear, and contempt. It is a useful dataset with its large size but accuracies of labels are low compared to facial recognition datasets due to the fact that it is created by web crawling face images with emotion-related keywords. In 2016, researchers from Windows decided to enhance this dataset by relabeling it which is called FER+.

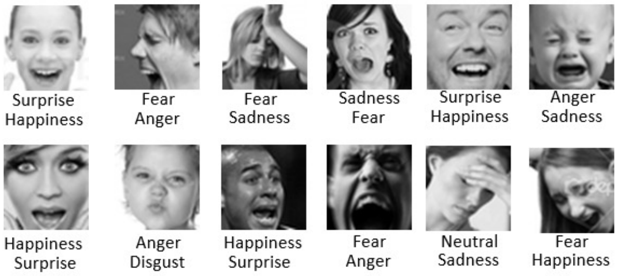

Figure 5. A few examples of emotions that were labeled incorrectly in the original FER. Labels at the top are from the original FER, below are the FER+

FER+ consists of 35.887 grayscale face crops which are labeled by 10 taggers. It has 10 labels which are 8 emotion types used in the original FER dataset, unknown and not a face. Since the FER dataset is created by web crawling, it has images that aren't much useful such as images that don't have a face or faces with an expression to hard to label. Extra two labels added in FER+ will help to handle these situations. While labeling dataset for FER+, taggers are asked to label the images and total answers achieved for every label in the image are stored in a file unlike the original FER dataset, which has a single emotion label for images.

4.3. Our Environment

We will use Google Colab service to train and evaluate our deep learning models. Google Colab is a free cloud service which has GPU & TPU support. The programming language we will use is Python3. We will benefit from various image processing, deep learning, data manipulation libraries such as numpy, OpenCV, etc.

5. Final Experimental Settings

We have used two different datasets. One is CK+ and the other is the FER+ dataset. We have started our project with the CK+ dataset but then we have experimented that the CK+ data set was insufficient and imbalanced. The preprocessing steps are different for these datasets. We have used CascadeClassifier from the OpenCV library to detect face regions in each frame. Because the frames in CK+ were not cropped for faces. Also, the image sizes were larger than 224, so we have resized the frames in size of 224x224 to be able to give them into the VGG-16 model. For the FER+ dataset, we did not use CascadeClassifier because the face regions were already cropped in this dataset. The image sizes were small, 48x48. So we have resized them to 224x224 to be able to give them into the VGG-16 model.

6. Experimental Results

6.1. CK+

CK+ dataset actually consists of video frames. But we have only used the last three images of the dataset. Also, we have created the ”neutral” emotion class from the first frames of each video. For other 7 emotion classes we have used the last 3 frames of the videos.

6.1.1 Extracting The Features with CNN

We have extracted the image features using the VGG-16 network. From VGG-16, we have used the layers until the pool5 layer and extracted the pool5 features for each image. Then we have used ”average pooling” to these features. Then we stored these features for each image in an array. Also, we have tried with ”max pooling”. Because the effects of these layers are different. Max pooling is used to extract the most important features like edges, takes the maximum. Average pooling extracts features smoothly, takes the average. But the consequences were not different than the ”average pooling”. To calculate the loss, we have used Cross Entropy Loss. Since in Cross Entropy Loss there exists a Softmax function, we did not use a Softmax function.

6.1.2 CNN - Fully Connected Layers

For the fully connected layers of our model, we have tried two different versions. First, we have used the VGG-16 network's fully connected layer by modifying the last fully connected layer to predict only 8 classes. Second, we have extracted the layers until the fully-connected layers from the VGG-16 network. Then we have created a Net class and in this class, we have implemented a forward function to build fully connected layers. From either of these approaches, we could not get good accuracy. When we have used fully-connected layers from the first approach, we have only predicted two classes which are ”neutral” and ”disgust”. The accuracy was 43.790849673202615 for the test dataset. Even tough the predictions were bad, the accuracy was considerably high due to the fact that our model mostly predicted ”neutral” which happens to be the most common type of data in our dataset. This result did not change even if we have tried to give a lower learning parameter. Also for each epoch, train loss has changed just a little. This made us think that we have not trained our model properly.

When we have applied fully connected layers from the second approach, we have predicted 5 classes at first. Then we have used a lower learning rate. The predicted class number became 5. We have tried adding dropout and our predicted class number became 6. But we have faced the same problem again. For each epoch, the loss has only changed a little. This little change shows that we could not train our model properly.

As we have said, CK+ is originally a video dataset. After we have created a ”neutral” emotion class by adding the first frames of each video, the number of images in this class became larger than other classes. This caused imbalanced data. So we believe that the training problem occurs because of this.

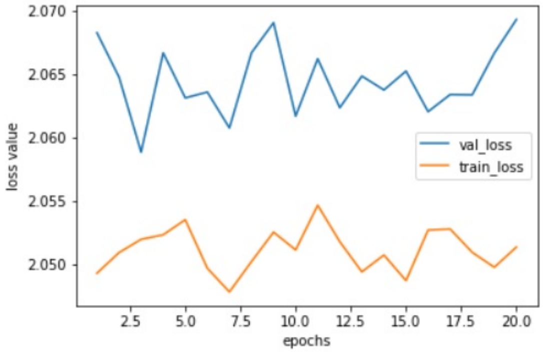

Figure 6. learning rate = 0.00001 & epoch = 20 batch size = 10

As we can see in the Figure 5, loss values are in a small range. Our model is not converging. We have tried other parameters as well but the result did not improve.

6.2. FER+

Since our dataset with CK+ was imbalanced and insufficient, we have decided to try our model with the FER+ dataset. FER+ dataset has 35.887 face crops. Each crop is in size of 48x48. Since the dataset is big, extracting the pool5 layer features took a very long time.

6.2.1 Fine-Tuning FC Layers

As we have done previously on the CK+ dataset, we have tried the VGG-16 network. Here, we have changed the last fully-connected layer output number as 10 since we have 10 emotion class in the FER+ dataset. We have trained our network with these pool5 layer features. We have started the training process with batch size 32, but we have got better results with batch size 1. So for the following, we have used batch size as 1. The following figures show the effect of the parameters for training the model.

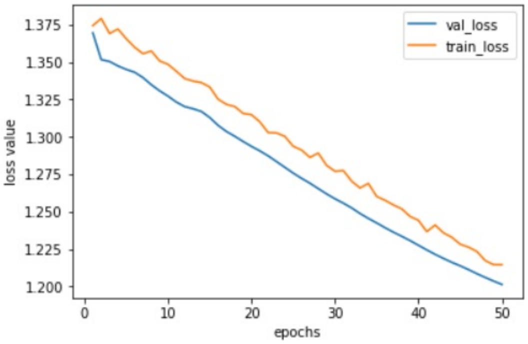

Figure 7. learning rate = 0.001 & epoch = 50

In Figure 7, we can see the result of using the learning rate as 0.001 and epoch as 50. This learning rate is too big for our model. Validation loss is under the train loss.

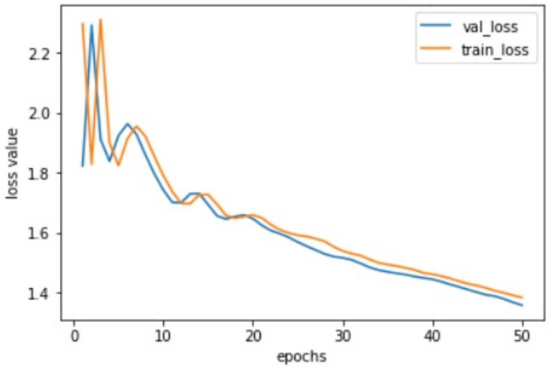

Figure 8. learning rate = 0.005 & epoch = 50

In Figure 8, we can see the result of using the learning rate as 0.005 and epoch as 50. This learning rate gave better convergence than the previous one. Validation loss is under the train loss but their line moves pretty similarly to each other.

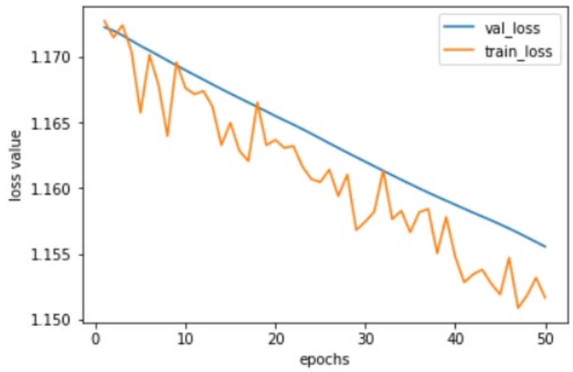

Figure 9. learning rate = 0.0001 & epoch = 50

In Figure 9, we can see the result of using the learning rate as 0.0001 and epoch as 50. Here train loss line is under the validation loss line which is the common case.

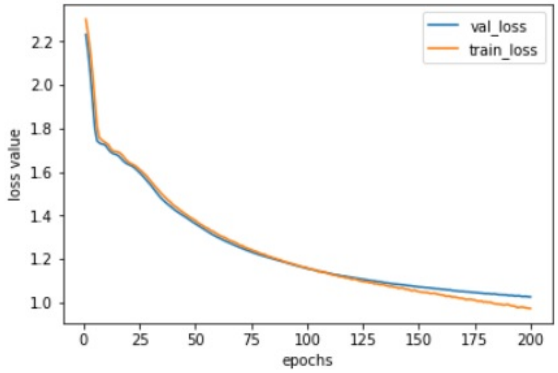

Figure 10. learning rate = 0.001 & epoch = 200

In Figure 10, we can see the result of using the learning rate as 0.001 and epoch as 200. The model is converging better than the model in Figure 6. But again the train loss line and validation loss line moves pretty similarly.

In Figure 11, we can see the result of using the learning rate as 0.005 and epoch as 200. Here we have trained the model better than the previous one.

In Figure 12, we can see the result of using the learning rate as 0.0001 and epoch as 200. This graph is quite similar to the previous one. But when we have printed each loss in each epoch, we can see that the train loss goes to 0.9. It might seem that this model gave the best result but if we look at Table 1, we can see that this model gave the lowest accuracy. As far as we understood, the reason behind is overfitting. It has a small learning rate and the number of epochs is really high. High number of epochs cause our model to fit better on train images but results in low accuracy on test images.

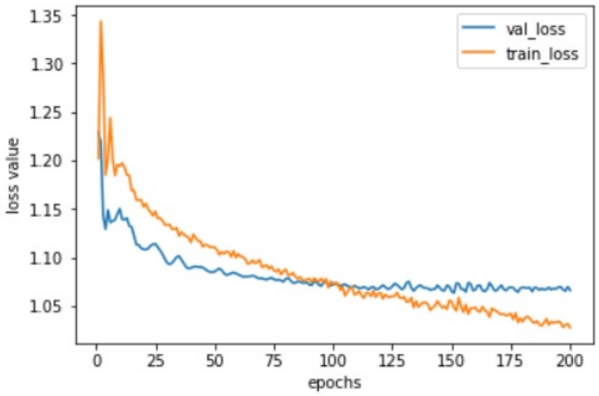

Figure 11. learning rate = 0.005 & epoch = 200

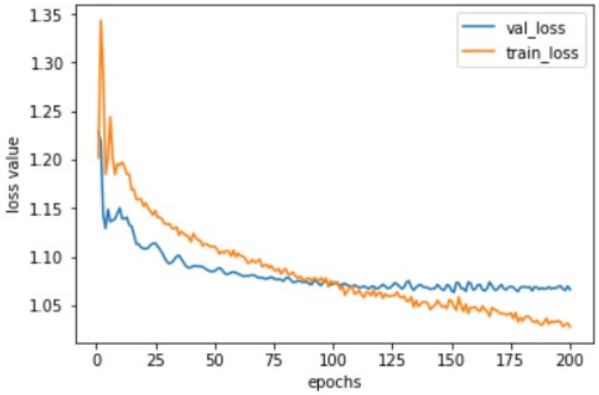

Figure 12. learning rate = 0.0001 & epoch = 200

From the understanding of Figure 11, we did not increase the epoch number. There is no need for that because the higher the epoch the more our model overfits.

Learning Rate	#Epochs	Accuracy	#Class
0.001	50	28.4914	5
0.005	50	31.4021	3
0.0001	50	28.3515	5
0.001	200	26.7842	6
0.005	200	26.8401	6
0.0001	200	26.2244	6

Table 1. Models Applied on Test Data

Table 1 shows the learning rate, number of epochs parameters and test dataset accuracies, number of classes that are predicted with respect to these parameters. If we consider only the accuracy, the second model gave the best result on test data. However, when we look at the number of classes it predicted, it is 3 which is quite insufficient. If we both consider the accuracy and number of predicted classes then we can see that the fifth model gave the best results on test data.

You can find the datasets and codes in this link.

References

[1] Emad Barsoum, Cha Zhang, Cristian Canton Ferrer, and Zhengyou Zhang. Training deep networks for facial expression recognition with crowd-sourced label distribution. In *ACM International Conference on Multimodal Interaction (ICMI)*, 2016.

[2] Peter Belhumeur, Joao Hespanha, and David Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19:711–720, 07 1997.

[3] Dan Duncan and Gautam Shine. Facial emotion recognition in real time. 2016.

[4] Jonathan Jonathan, Andreas Lim, Paoline , I Gede Putra Kusuma Negara, and Amalia Zahra. Facial emotion recognition using computer vision. pages 46–50, 09 2018.

[5] Takeo Kanade, Jeffrey Cohn, and Yingli Tian. Comprehensive database for facial expression analysis. *4th IEEE International Conference on Automatic Face and Gesture Recognition, France*, 02 1970.

[6] Byoungchul Ko. A brief review of facial emotion recognition based on visual information. *Sensors*, 18:401, 01 2018.

[7] P. Lucey, J. Cohn, S. Lucey, I. Matthews, S. Sridharan, and K. M. Prkachin. Automatically detecting pain using facial actions. In *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, pages 1–8, Sep. 2009.

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

[9] Kenneth Prkachin. The consistency of facial expressions of pain: A comparison across modalities. *Pain*, 51:297–306, 01 1993.

[10] Andrew Ryan, Jeffrey Cohn, Simon Lucey, Jason Saragih, Patrick Lucey, Fernando De la Torre, and Adam Rossi. Automated facial expression recognition system. pages 172 – 177, 11 2009.

[11] Marian Stewart Bartlett, Gwen Littlewort, Ian Fasel, and Javier Movellan. Real time face detection and facial expression recognition: Development and applications to human computer interaction. volume 5, pages 53–53, 07 2003.

[12] Pawe Tarnowski, Marcin Koodziej, Andrzej Majkowski, and Remigiusz Rak. Emotion recognition using facial expressions. *Procedia Computer Science*, 108:1175–1184, 12 2017.

[13] Alessandro Vinciarelli, Maja Pantic, and Herve Bourlard. Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 27:1743–1759, 11 2009.

[14] Esra Vural, Mujdat Cetin, Aytul Ercil, Gwen Littlewort, Marian Bartlett, and Javier Movellan. Automated drowsiness detection for improved driving safety. 05 2019.