Emotion Recognition From Facial Expressions: A Preliminary Report

Tanja Jaschkowitz* Leah Kawka* Mahdi Mohammadi* Jiawen Wang*
{Tanja.Jaschkowitz,Leah.Kawka,Mahdi.Mohammadi,Jiawen.Wang}@campus.lmu.de

1. Introduction

Facial emotion recognition (FER) [1] is not only an interesting in our daily life, but also important in the realm of artificial intelligence and computer vision. In this short proposal, we aim to leverage several deep neutral networks to analyze and interpret different human facial emotions.

The structure of this report is arranged as follows. In Section 2, we provide the datasets we used, the model architecture we implemented, and preliminary evaluation results of our model. Our code and supplementary material is available at https://github.com/werywjw/SEP-CVDL.

2. Approach

2.1. Dataset Aquisition and Processing

Firstly, for all the image data from the training dataset RAF-DB[1] [2, 3], we filter out neutral instances from the original dataset, the emotion labels are denoted as 1 (Surprised), 2 (Fearful), 3 (Disgusted), 4 (Happy), 5 (Sad), and 6 (Angry) for simplicity (Our first dataset is downloaded from https://www.kaggle.com/datasets/shuvoalok/raf-db-dataset/code with the addition csv file to their labels). The test result in Figure 2 is also aggregated from this specific dataset. To transform and resize the images to `(64,64)`, we convert the images to greyscale with three channels as our original CNN is designed to work with three-channel inputs. Also, we randomly flip the images horizontally with a default 50% chance. This kind of augumentation helps in making the model more robust to orientation changes and thus improve the generalization ability. Our training dataset is aggregated from CK+ [4].

2.2. Model Architecture

We implemented from scratch an emotion-classification model with four convolution layers at the very begining. Following each convolutional layer, batch normalization is applied. This stabilizes learning by normalizing the input to each layer. Then three linear layers are applied to extract features to the final output. We also add a `dropout` layer to prevent overfitting. The activation function used after each layer is Rectified Linear Unit (ReLU), since it introduces the non-linearity into the model, allowing it to learn more complex patterns. In order to find the best hyperparameter configuration (see Tab. 2 for details) of the model, we utilize the parameter grid from sklearn [2].

Models	Accuracy (Vali)	Accuracy (Test)	Accuracy (Train)
Baseline CNN	66.3	75.2	52.6
ResNet18	76.8	79.8	60.3

Table 1. Accuracy (%) for different models in our experiments

Hyperparameter	Configuration
Learning rate	{0.1, 0.01, 0.001, 0.0001}
Batch size	{8, 16, 32, 64}
Dropout rate	{0.5}
Epoch	{10, 20, 30}
Early stopping	{`True`, `False`}
Patience	{5}

Table 2. Explored hyperparameter space for our model

2.3. Preliminary Results

For evaluation, we use the metric accuracy. The loss function is cross entropy, which is typically for multi-class classification. The main results of our experiment with respect to loss and accuracy is depicted in Figure 2.

3. Optimization Strategies

In the coming weeks, we

Author Contributions

Equal contributions listed by alphabetical order of surnames. Every author did the literature research and contributed to the writing of the paper. An overview of our time schedule for the entire final project is given in Figure 1.

[1] http://www.whdeng.cn/raf/model1.html

[2] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.ParameterGrid.html

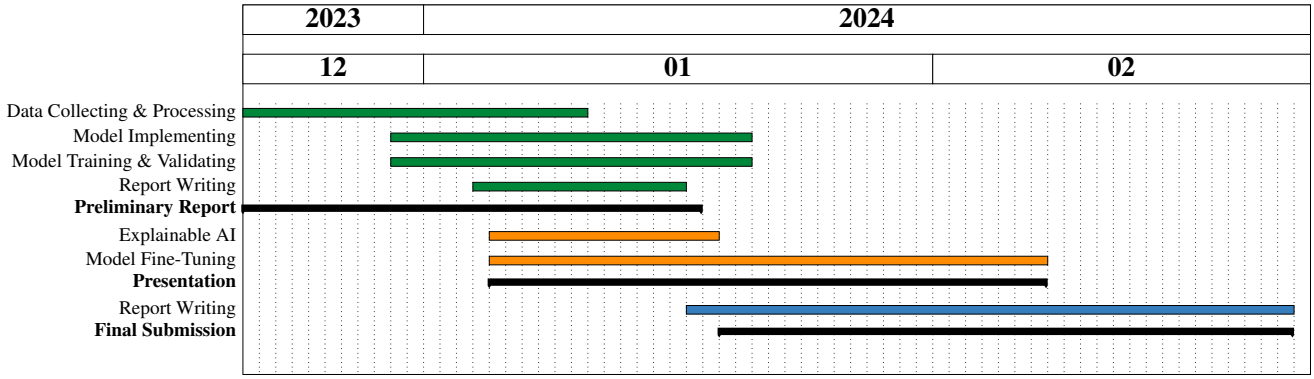

Figure 1. Overview of the time schedule for the final project

Figure 2. Empirical results in terms of the loss and accuracy on differen training epochs

- **Tanja Jaschkowitz** implemented the model architecture, training and testing infrastructure,
- **Leah Kawka** collected the training data, prepared dataprocessing, implemented augmentation, run the results, Explainable AI & Video-green square
- **Mahdi Mohammadi** implemented the
- **Jiawen Wang** implemented the model architecture, training and testing infrastructure, and optimization strategies. In the specific wrting part, she also checked and aggereated this report from other team members.

Acknowledgements

We are deeply grateful to our advisors **Johannes Fischer** and **Ming Gui** for their helpful and valuable support during the entire semester. We also thank **Prof. Dr. Björn Ommer** for providing this interesting practical course.

References

[1] ByoungChul Ko. A brief review of facial emotion recognition based on visual information. *Sensors*, 18(2):401, 2018. 1

[2] Shan Li and Weihong Deng. Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition. *IEEE Transactions on Image Processing*, 28 (1):356–370, 2019. 1

[3] Shan Li, Weihong Deng, and JunPing Du. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2584–2593. IEEE, 2017. 1

[4] Patrick Lucey, Jeffrey F. Cohn, Takeo Kanade, Jason M. Saragih, Zara Ambadar, and Iain A. Matthews. The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2010, San Francisco, CA, USA, 13-18 June, 2010*, pages 94–101. IEEE Computer Society, 2010. 1