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) [4, 5] is a topic of significant frontier and ongoing debate, not only in our daily life, but also in the fields of *artificial intelligence* (AI) and computer vision. In this short proposal, we aim to leverage several *deep neural networks* (DNNs), which contain convolution layers and residual/attention blocks, to detect and interpret six basic universally recognized and expressed human facial emotions (i.e., happiness, surprise, sadness, anger, disgust, and fear). To make our model more transparent, we explain this emotion classification task with *class activation mapping* (CAM) and *gradient-weighted class activation mapping* (Grad-CAM).

The structure of this report is arranged as follows. In Section 2, we address the datasets we collected and the model architecture we implemented. The preliminary evaluation results of our models are given in Section 3. Section 4 describes the optimization strategies we have plan to investigate in the coming weeks. Figure 1 illustrates the empirical results of our current best model, an overview of our time schedule for the entire final project is given in Figure 2. Our code and supplementary material are available at `https://github.com/werywjw/SEP-CVDL`.

2. Approach

2.1. Dataset Acquisition and Processing

To initiate the project, we acquired the databases such as FER+ [1] [1], RAF-DB [6, 7], CK+ [9], TFEID [2, 8], as well as the video database DISFA [10], from public institutions and GitHub repositories [2]. Based on these databases, we created a dataset by augmentation to increase variety, full details of augmentation (see Section 4.1 for details). In terms of illustrating the content of used pictures, we exclusively analyze human faces representing 6 emotions. That is, we generalized a folder structure annotating the labels 1 (surprise), 2 (fear), 3 (disgust), 4 (happiness), 5 (sadness), and 6 (anger). Besides the original format of images and videos, we set standards for extracting frames from the videos, resize training pictures to 64x64 pixels, and save them as the JPG format.

The images are converted to greyscale with three channels, as our original *convolutional neural network* (CNN) is designed to work with three-channel inputs with random rotation and crop. Emotions were assigned tags to each individual picture in a CSV file to facilitate further processing in the model. We create a custom dataset, which is a collection of data relating to all training images we collected, using PyTorch [3], as it includes plenty existing functions to load various custom datasets in domain libraries such as `TorchVision`, `TorchText`, `TorchAudio`, and `TorchRec`.

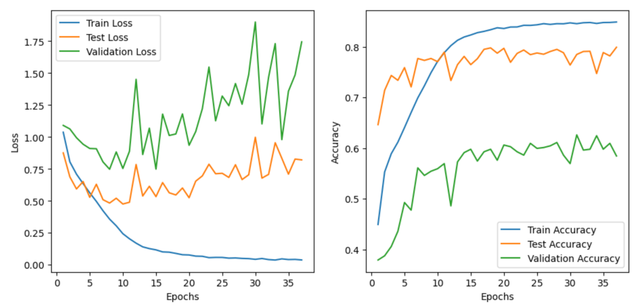

Figure 1. Empirical results in terms of the loss and accuracy with respect to different epochs for our current best model

2.2. Model Architecture

We implement an emotion classification model from scratch with four convolution layers at the very beginning. Following each convolutional layer, batch normalization is used, as this stabilizes learning by normalizing the input to each layer. Afterward, three linear layers are applied to extract features to the final output. We also add a dropout layer to prevent overfitting. The activation function after each layer is *Rectified Linear Unit* (ReLU), since it introduces the nonlinearity into the model, allowing it to learn more complex patterns.

In order to find the best hyperparameter configuration (see Table 1 for details) of the model, we utilize the pa-

[1] `https://github.com/microsoft/FERPlus`
[2] `https://github.com/spenceryee/CS229`
[3] `https://pytorch.org`

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, 40}
Early stopping	{`True`, `False`}
Patience	{5}

Table 1. Explored hyperparameter space for our models

rameter grid from sklearn [4]. Additionally, we increase the depth of the network by adding some convolutional layers to learn more complex features. To help the training of deeper networks more efficiently, we add the residual connections, as they allow gradients to flow through the network more easily, improving the training for deep architectures. Moreover, we add squeeze and excitation (SE) blocks to apply channel-wise attention.

3. Preliminary Results

For evaluation, we use the metric accuracy. The loss function employed for all models is cross-entropy, which is typically for multi-class classification. We report all the training, testing, and validation accuracy in % to compare the performance of our models.

Figure 1 shows the test result aggregated from the database RAF-DB [5]. As seen in Table 2, our CNN without random augmentation outperforms the other models in terms of the accuracy, indicating that this kind of augmentation is not able to help our model predict the correct the label, thus we later aim to optimize with other augmentation techniques to capture more representative features of different emotions. Further research is orientated on papers engaging similar investigations [7, 12, 13].

Models	Accuracy (Train)	Accuracy (Test)	Accuracy (Vali)
CNN (Baseline)	66.3	75.2	52.6
ResNet18 [3]	76.8	79.8	60.3
CNN (SE+Aug)	74.3	79.9	59.6
CNN (SE-Aug)	84.6	79.5	**62.8**
CNN (SE+Res)	71.5	78.9	56.4
CNN (Ours-Aug)	**84.9**	**80.0**	62.6

Table 2. Accuracy (%) for different models in our experiments (Note that Aug stands for data augmentation, SE for squeeze and excitation, and Res for residual connections; +/- represent with/without respectively.)

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

[5] https://www.kaggle.com/datasets/shuvoalok/raf-db-dataset

4. Optimization Strategies

4.1. Data Augmentation

In machine learning and AI, augmentation stands as a transformative technique, empowering algorithms to learn from and adapt to a wider range of data. By introducing subtle modifications to existing data points, augmentation effectively expands the dataset, enabling models to generalize better and achieve enhanced performance. As models encounter slightly altered versions of familiar data, they are forced to make more nuanced and robust predictions. With this process, we aim to prevent overfitting. Additionally, we guide the training process to enhance the recognition and handling of real-world variations. During the project, we pursue various approaches. We are implementing different combinations of functions from the `pytorch.transforms` library and testing already established filters that have been developed. Meanwhile, we create various replications of existing photos by randomly altering different properties such as size, brightness, color channels, or perspectives.

4.2. CAM

Generally speaking, Class Activation Mapping is a visualization technique designed to highlight the regions of an image or video that contribute the most to the prediction of a specific class by a neural network, typically the final convolutional layer of a CNN before the fully connected layers. Technically, CAM generates a heatmap that highlights the important regions of the image in terms of the decision of the model. Besides proposing a method to visualize the discriminative regions of a classification-trained CNN, we adapte this approach from Zhou et al. [14] to localize objects without providing the model with any bounding box annotations. The model can thus learn the classification task with class labels and is then able to localize the object of a specific class in an image.

The final convolutional layer produces feature maps, and the GAP layer computes the average value of each feature map. The weights connecting the feature maps to the output class are obtained. The weighted combination of feature maps, representing the importance of each spatial location, is used to generate the CAM heatmap. CAM helps interpret CNN decisions by providing visual cues about the regions that influenced the classification. It aids in understanding the model's behavior and can be useful for model debugging and improvement. The global average pooling (GAP) layer is used to obtain a spatial average of the feature maps.

4.3. Grad-CAM

Despite CAM can provide valuable insights into the decision-making process of deep learning models, especially CNNs, CAM must be implemented in the last layer of

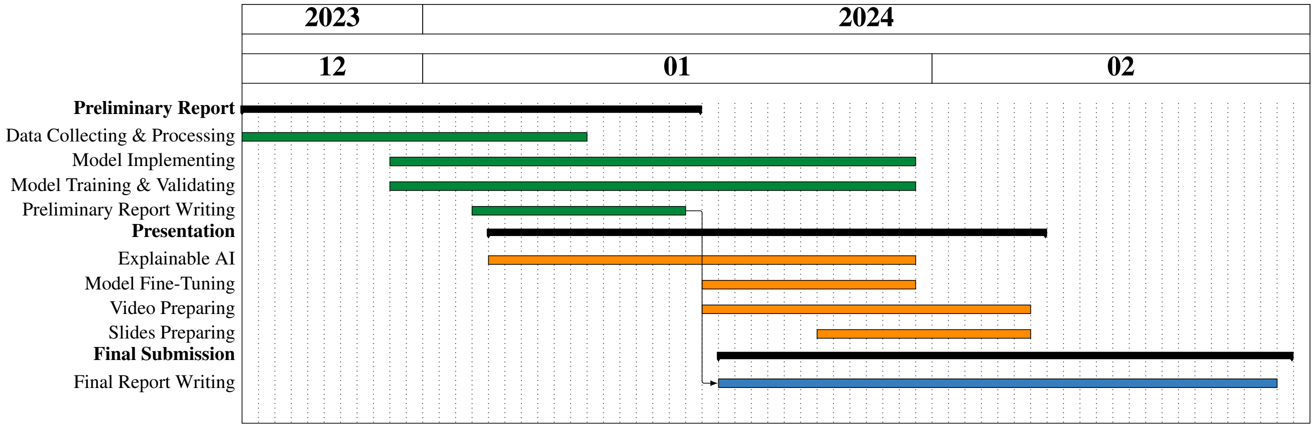

Figure 2. Overview of the schedule for the final project

a CNN, We thus follow up Gradient-weighted CAM [11], introduced as a technique that is easier to implement with different architectures. This task will be implemented by using the libraries of Pytorch and OpenCV [6].

4.4. Table of Classification scores

To further analyze the separate scores of the each class of the model, we wrote a script that takes a folder path as input and iterates through the images inside a subfolder. The output is a CSV file representing the corresponding classification scores.

Author Contributions

Equal contributions are listed by alphabetical order of surnames. Every author did the literature research and contributed to the writing of the paper.

- **Tanja Jaschkowitz** implemented the model architecture, training and testing infrastructure, and CSV file aggregations.
- **Leah Kawka** collected the training data, prepared data processing, implemented augmentation, and ran the results. She also takes part in the explainable AI and Grad-CAM.
- **Mahdi Mohammadi** implemented the augmentation, did the research searching, conclusion reasearching, data preprocessing, and CAM-Images inquiry.
- **Jiawen Wang** implemented the model architecture, training and testing infrastructure, and optimization strategies. In the specific writing part, she also draw the figures and tables and improved 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.

[6] https://opencv.org

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 *Proceedings of the 18th ACM International Conference on Multimodal Interaction, ICMI 2016, Tokyo, Japan, November 12-16, 2016*, pages 279–283. ACM, 2016. 1

[2] L.F. Chen and Y.S. Yen. Taiwanese facial expression image database, 2007. 1

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. 2

[4] Deepak Kumar Jain, Pourya Shamsolmoali, and Paramjit S. Sehdev. Extended deep neural network for facial emotion recognition. *Pattern Recognit. Lett.*, 120:69–74, 2019. 1

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

[6] 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

[7] 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 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2584–2593. IEEE Computer Society, 2017. 1, 2

[8] Shanshan Li, Liang Guo, and Jianya Liu. Towards east asian facial expression recognition in the real world: A new database and deep recognition baseline. *Sensors*, 22(21): 8089, 2022. 1

[9] 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

[10] Seyed Mohammad Mavadati, Mohammad H. Mahoor, Kevin Bartlett, Philip Trinh, and Jeffrey F. Cohn. DISFA: A spontaneous facial action intensity database. *IEEE Trans. Affect. Comput.*, 4(2):151–160, 2013. 1

[11] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 618–626. IEEE Computer Society, 2017. 3

[12] Monu Verma, Murari Mandal, M. Satish Kumar Reddy, Yashwanth Reddy Meedimale, and Santosh Kumar Vipparthi. Efficient neural architecture search for emotion recognition. *Expert Syst. Appl.*, 224:119957, 2023. 2

[13] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, pages 818–833. Springer, 2014. 2

[14] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2921–2929. IEEE Computer Society, 2016. 2