Emotion Recognition From Facial Expressions

Tanja Jaschkowitz* Leah Kawka* Mahdi Mohammadi* Jiawen Wang*

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

Abstract

Facial emotion recognition (FER) [6, 7] is a topic of significant frontier and ongoing debate, not only in our daily lives but also in the fields of artificial intelligence (AI) and computer vision. However, the existing FER approaches are not always reliable or explainable, we here propose our model with explainable AI, i.e., via layer activations, CAM, and Grad-CAM. Empirical results show that our model can achieve comparable state-of-the-art performance in terms of the accuracy. Our code and supplementary material are available at `https://github.com/werywjw/SEP-CVDL`.

1. Introduction

Facial emotion recognition (FER) [6, 7] is a topic of significant frontier and ongoing debate, not only in our daily lives but also in the fields of *artificial intelligence* (AI) and computer vision. In this report, 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).

Our main contributions can be summarized as follows.

- We collect and preprocess the training and testing data (image and video) from various public databases.
- We implement all CNN models from scratch and optimize it with different augmentation techniques. Meanwhile, we provide the classification scores of each emotion class in a comprehensive CSV file with respect to each image.
- We give the video demo to illustrate the real-world performance of our best model.
- We provide qualitative benefits such as interpretability to explain our model with Grad-CAM.

The structure of this report is arranged as follows. Section 2 contains the related work of our research. In Section 3, we address the datasets we collected and the model architecture we implemented. The preliminary evaluation results of our models are given in Section 4.4. Section 5 describes the optimization strategies we plan to investigate in the coming weeks. Figure 2 illustrates the empirical results of our current best model.

2. Related Work

2.1. Explainable AI

3. Approach

4. Evaluation

4.1. Experimental Setup

4.2. Dataset Description

To initiate the project, we acquired the databases such as RAF-DB [8, 9], FER+ [1], CK+ [11], TFEID [3, 10], as well as the video database DISFA [12], from public institutions and GitHub repositories [14]. Based on these databases, we created a dataset by augmentation to increase the variety, and full details of augmentation (see Section 5.1). 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 0 (happiness), 1 (surprise), 2 (sadness), 3 (anger), 4 (disgust), and 5 (fear). Besides the original format of images and videos, we set standards for extracting frames from the videos, resizing training pictures to 64×64 pixels, and saving them in 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, as it includes plenty of existing functions to load various custom datasets in domain libraries such as `TorchVision`, `TorchText`, `TorchAudio`, and `TorchRec`.

Dataset	#Images in training set	#Images in testing set	# Images in validation set
Real-world Affective Faces Database (RAF-DB [8, 9])	9747	2388	600
Facial Expression Recognition Plus (FER+ [1])	23743	5945	600
Extended Cohn-Kanade Dataset Plus (CK+ [11])		-	600
Taiwanese Facial Expression Image Database (TFEID [3, 10])		-	600

Table 1. Overview of the datasets statistics used in our experiment

4.3. Model Architecture

The input to our emotion recognition model is a gray scale image at 64×64 resolution. The output is 6 emotion classes: happiness, surprise, sadness, anger, disgust, and fear. We implement an emotion classification model from scratch with four convolution layers at the very beginning. Following each convolutional layer, batch normalization and interleaved with max pooling are 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 50% dropout layer to prevent overfitting. Verified by Barsoum et al. [1], the dropout layers are effective in avoiding model overfitting. The activation function 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 Table 3 for details) of the model, we utilize the parameter grid from sklearn. 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.

4.4. Results

For evaluation, we use the metric accuracy. The loss function employed for all models is cross-entropy (CE), which is typically for multi-class classification. That is:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{n} y_i \log(p_i), \tag{1}$$

where y_i is the true label and p_i is the predicted probability of the i-th class. We report all the training, testing, and validation accuracy in % to compare the performance of our models.

Adding a extra convolutional layer to the model with much more parameters does not necessarily lead to better performance. Batch normalization can indeed improve the performance of the model.

Figure 1 shows the test result aggregated from the database RAF-DB [4]. Different combinations of functions from the `pytorch.transforms` library are tested for augmentation from those already established filters. As seen in Table 4, our CNN without random augmentation outperforms the other models in terms of accuracy, indicating that this kind of augmentation is not able to help our model predict the correct 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 [9, 15, 16].

5. Optimization Strategies

5.1. Data Augmentation

In deep 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. Meanwhile, we create various replications of existing photos by randomly altering different properties such as size, brightness, color channels, or perspectives.

5.2. Classification Scores

To further analyze the separate scores of each class of the model, we write a script that takes a folder path as input and iterates through the images inside a subfolder to record the performance of the model with respect to each emotion class. This CSV file is represented with the corresponding classification scores.

5.3. CAM and Grad-CAM

In generally, CAM [17] helps interpret CNN decisions by providing visual cues about the regions that influenced the classification, as it highlights the important regions of an image or a video, aiding in the understanding of the behavior of the model, which is especially useful for model debugging and improvement. Besides proposing a method

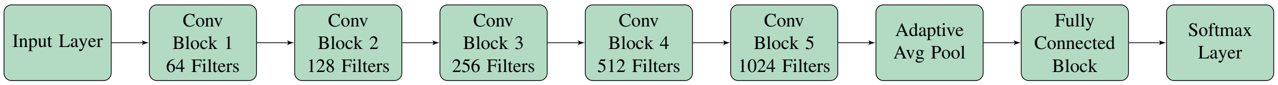

Figure 1. Overview of the model architecture (see Figure 2 for a detailed version)

filepath	happiness	surprise	sadness	anger	disgust	fear
archive/RAF-DB/test/test_0524_aligned_happiness.jpg	**1.00**	0.00	0.00	0.00	0.00	0.00
archive/RAF-DB/test/test_0093_aligned_happiness.jpg	**1.00**	0.00	0.00	0.00	0.00	0.00
archive/RAF-DB/test/test_2193_aligned_sadness.jpg	0.03	0.01	**0.90**	0.01	0.02	0.04
archive/RAF-DB/test/test_1214_aligned_happiness.jpg	**1.00**	0.00	0.00	0.00	0.00	0.00
archive/RAF-DB/test/test_1816_aligned_surprise.jpg	0.00	**1.00**	0.00	0.00	0.00	0.00
archive/RAF-DB/test/test_0294_aligned_surprise.jpg	0.01	**0.99**	0.00	0.00	0.00	0.00
archive/RAF-DB/test/test_1128_aligned_happiness.jpg	**1.00**	0.00	0.00	0.00	0.00	0.00
archive/RAF-DB/test/test_1799_aligned_sadness.jpg	0.37	0.02	**0.45**	0.02	0.1	0.03
archive/RAF-DB/test/test_0610_aligned_sadness.jpg	0.02	0.00	**0.74**	0.02	0.21	0.01
archive/RAF-DB/test/test_1373_aligned_anger.jpg	0.00	0.00	0.00	**1.00**	0.00	0.00
archive/RAF-DB/test/test_1788_aligned_fear.jpg	0.00	0.00	0.00	0.00	0.00	**1.00**
archive/RAF-DB/test/test_0007_aligned_disgust.jpg	0.03	0.05	0.03	**0.58**	0.18	0.13
archive/RAF-DB/test/test_0804_aligned_disgust.jpg	0.02	0.00	0.18	0.02	**0.77**	0.00

Table 2. Overview of our random testing results examples extracted from the CSV file

Hyperparameter	Value
Learning rate	{0.01, 0.001, 0.0001}
Batch size	{8, 16, 32, 64}
Dropout rate	{0.2, 0.5}
Epoch	{20, 30, 40}
Early stopping	{`True`, `False`}
Patience	{5}

Table 3. Explored hyperparameter space for our models

to visualize the discriminative regions of a CNN trained for the classification task, we adopt this approach from Zhou et al. [17] to localize objects without providing the model with any bounding box annotations. The model can therefore learn the classification task with class labels and is then able to localize the object of a specific class in an image or video.

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 a CNN or before the fully connected layer, We will meanwhile compare to Gradient-weighted CAM [13], introduced as a technique that is easier to implement with different architectures. This task will be implemented by using the libraries from PyTorch and OpenCV [1].

Chattopadhay et al. [2] proposed Grad-CAM++,

[1] https://opencv.org

6. Conclusion and Discussion

7. Limitation

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, classification score script, and optimization strategies. In the specific writing part, she drew the figures and tables and improved this report from 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.

Models	Architecture	Training Accuracy	Testing Accuracy	Validation Accuracy	# Parameters
CNN (4 layers)	+BN-SE	96.6	80.6	66.8	2606086
ResNet18 [5]	residual block	98.9	81.3	67.9	11179590
CNN (5 layers)	-BN-SE	96.3	76.9	60.6	10474118
CNN (5 layers)	+BN+SE	98.4	81.7	71.1	10478598
CNN (5 layers)	+BN-SE	98.6	83.1	72.1	10478086
CNN (6 layers)	+BN-SE	97.5	82.5	70.0	41950726
CNN (4 layers)	+BN-SE	86.6	64.1	40.2	2606086
CNN (5 layers)	+BN-SE	89.6	65.6	40.7	10478086
CNN (6 layers)	+BN-SE	96.0	65.5	41.6	41950726

Table 4. 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)

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

[2] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Improved visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018. 3

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

[4] Dev-ShuvoAlok. RAF-DB DATASET, https://www.kaggle.com/datasets/shuvoalok/raf-db-dataset, 2023. 2

[5] 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. 4

[6] 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

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

[8] 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, 2

[9] 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

[10] 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, 2

[11] 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, 2

[12] 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

[13] 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

[14] Spenceryee. CS229, https://github.com/spenceryee/CS229, 2016. 1

[15] 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

[16] 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

[17] 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, 3

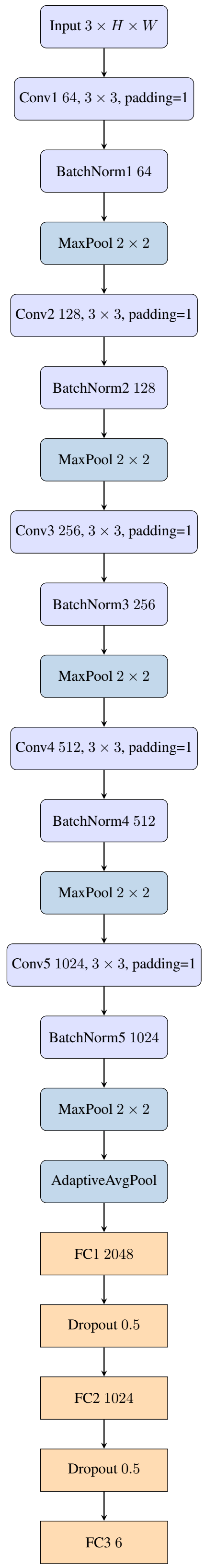

Figure 2. Overview of the detailed model architecture