GIMEFIVE: Towards Interpretable Facial Emotion Recognitions

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

Abstract

Facial emotion recognition 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 detection approaches are not always reliable or explainable, we here propose our model with interpretations, i.e., via layer activations, CAM, and Grad-CAM. Empirical results on facial emotion benchmarks show that our model achieves comparable state-of-the-art performance in terms of 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 *gradient-weighted class activation mapping* (Grad-CAM).

Our main contributions can be summarized as follows.

- We collect, preprocess, and evaluate the training and testing data throughly, (both image and video) from various public databases.
- We implement all classification models from scratch and optimize them with several techniques in a systematical manner. Meanwhile, we provide the classification scores of each emotion class in a detailed script 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.

An overview of the experimental pipeline of our project is illustrated in Figure 1. The structure of rest of the 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 evaluation results of our models are given in Section 4 with interpretability. Section 5 describes the optimization strategies such as data augmentation and hyper-parameter tuning. We provide the conclusion and discussion in Section 6.

2. Related Work

2.1. Facial Emotion Recognition

2.2. Explainable AI

To understand the decision-making process of our model, we aim to explain our model in a more transparent and interpretable way using the Grad-CAM, i.e., Gradient-weighted CAM [17], a technique that is easier to implement with different architectures.

Class Activation Mapping (CAM). CAM is a technique popularly used in CNNs to visualize and understand the regions of an input image that contribute the most to a particular class prediction. Generally speaking, CAM [20] 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 further improvement. Typically, CAM is applied to the final convolutional layer of a CNN. Besides proposing a method to visualize the discriminative regions of a CNN trained for the classification task, we adopt this approach from Zhou et al. [20] 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.

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

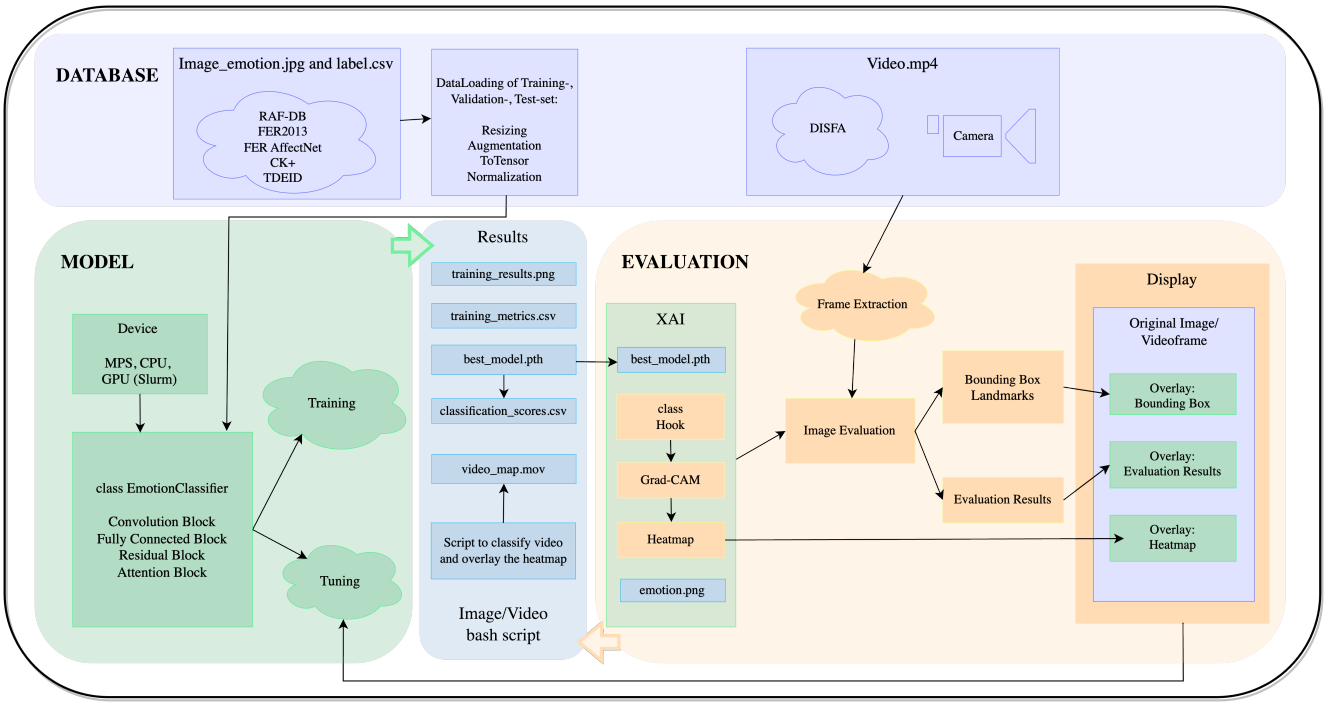

Figure 1. Overview the experimental pipeline of our project

3. Experimental Setup

All the experiments are implemented in Python and Shell for generating scripts.

3.1. Dataset Description

To initiate the project, we acquired the image databases such as RAF-DB [8, 9], FER+ [1], FER AffectNet [14], CK+ [11], and TFEID [3, 10], as well as the video database DISFA [12] from public institutions and kaggle [4, 16] 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.

3.2. Model Architecture

Figure 2 illustrates the overview of our model architecture. The input of our emotion recognition model is an image with 3 channels 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 blocks at the very beginning. Despite larger kernel can provide more information and wider area view due to more parameters, we use a 3×3 kernel size for all convolutional layers, as it is efficient to train and shares the weights without expensive computation. Following each convolutional layer, batch normalization is used for stabilizing the learning by normalizing the input to each layer. We interleaved with the max pooling layer because it reduces the spatial dimensions of the input volume. 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 4 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

DATASET	#Images in training set	#Images in testing set	# Video/Images in validation set
Denver Intensity of Spontaneous Facial Action Database (DISFA [13])	-	27	-
Real-world Affective Faces Database (RAF-DB [8, 9])	9747	2388	600
Facial Expression Recognition 2013 (FER2013 [1])	23743	5945	600
Fer AffectNet Database (FER AffectNet [14])	21045	-	600
Extended Cohn-Kanade Dataset Plus (CK+ [11])	309	-	600
Taiwanese Facial Expression Image Database (TFEID [3, 10])	229	-	600
FER GIMEFIVE	55073	8333	600

Table 1. Overview of statistics of datasets used in our experiment (Note that 600 images in the validation set above is given from our advisors)

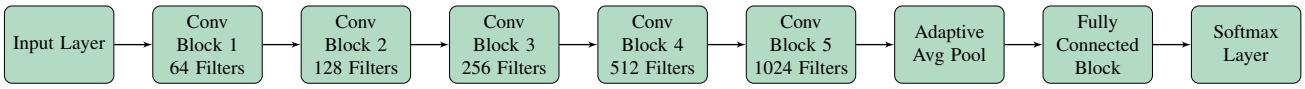

Figure 2. Overview of the model architecture (see Figure 3 for a detailed version)

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. Evaluation

For evaluation, we use the metric accuracy. We report all the training, testing, and validation accuracy in % to compare the performance of our models. 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.

4.1. Evaluation Results

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

Table 2 shows the test result aggregated from the database RAF-DB [4] and FER2013 [15]. Different combinations of functions from the `pytorch.transforms` library are tested for augmentation from those already established filters. As seen in Table 2, 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, 18, 19].

4.2. Interpretable Results

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.

In our case, we leverage the fifth convolutional layer of our model to generate the CAM heatmap. The *global average pooling* (GAP) layer, which computes the average value of each feature map to obtain a spatial average of feature maps, is used to obtain a spatial average of the feature maps.

5. Optimization Strategies

To further understand and enhance the performance of the model during training,

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, which is a common pitfall in machine learning. Additionally, we guide the training process to enhance the recognition and handling of real-world variations. Meanwhile, we create various replications of existing photos by randomly altering different properties such as size, brightness, color channels, or perspectives.

Dataset	Models	Architecture	Accuracies			# Parameters
			Training	Testing	Validation	
RAF-DB [8, 9]	ResNet18 [5]	Residual Block	**98.9**	81.3	67.9	11179590
	Ours (13 layers)	+BN-SE	96.6	80.6	66.8	2606086
	Ours (10 layers)	-BN-SE	96.3	76.9	60.6	10474118
	Ours (16 layers)	+BN+SE	98.4	81.7	71.1	10478598
	Ours (15 layers)	+BN-SE	98.6	**83.1**	**72.1**	10478086
	Ours (17 layers)	+BN-SE	97.5	82.5	70.0	41950726
FER2013 [1]	Ours (13 layers)	+BN-SE	86.6	64.1	40.2	2606086
	Ours (15 layers)	+BN-SE	89.6	**65.6**	40.7	10478086
	Ours (17 layers)	+BN-SE	**96.0**	65.5	**41.6**	41950726

Table 2. Accuracies (%) for different models (with specific architectures and numbers of parameters) in our experiments (Note that SE stands for the squeeze and excitation block and BN for the batch normalization; +/- represent with/without respectively)

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 3. 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}
Convolution depth	{4, 5, 6}
Fully connected depth	{2, 3}
Batch normalization	{`True`, `False`}
Pooling	{`max`, `adaptive avg`}
Optimizer	{`Adam`, `SGD`}
Activation	{ `relu`, `tanh`, `elu`, `gelu`}
Epoch	{30, 40, 50, 70, 100}
Early stopping	{`True`, `False`}
Patience	{5, 10}

Table 4. Explored hyperparameter space for our models

5.2. Hyper-parameter Tuning

6. Conclusion and Discussion

7. Limitation

Author Contributions

- **Jiawen Wang** implemented different model architectures from scratch, training and testing infrastructure, classification score script, Grad-CAM explanation, and optimization strategies, together with the corresponding writing part.
- **Leah Kawka** collected the training data, prepared data processing, and implemented augmentation. She helps run the results with Slurm. She also takes part in the explainable AI and Grad-CAM. In the specific writing part, she drew the
- **Mahdi Mohammadi** implemented the augmentation, did the research searching, conclusion researching, data pre-

processing, and CAM-Images inquiry.

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. We especially acknowledge our former team member **Tanja Jaschkowitz**, who joined us during the first phase of the project and shared two notebook scripts.

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. 2, 3, 4

[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. 1

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

[4] Dev-ShuvoAlok. RAF-DB DATASET: For recognize emotion from facial expression, https://www.kaggle.com/datasets/shuvoalok/raf-db-dataset, 2023. 2, 3

[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. 2, 3, 4

[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. 2, 3, 4

[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. 2, 3

[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. 2, 3

[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. 2

[13] S Mohammad Mavadati, Mohammad H Mahoor, Kevin Bartlett, Philip Trinh, and Jeffrey F Cohn. Disfa: A spontaneous facial action intensity database. *IEEE Transactions on Affective Computing*, 4(2):151–160, 2013. 3

[14] Ali Mollahosseini, Behzad Hasani, and Mohammad H. Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1):18–31, 2019. 2, 3

[15] Manas Sambare. Fer-2013: Learn facial expressions from an image, https://www.kaggle.com/datasets/msambare/fer2013, 2020. 3

[16] Noam Segal. Facial expressions training data: Fer affectnet database, https://www.kaggle.com/datasets/noamsegal/affectnet-training-data/data, 2022. 2

[17] 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. 1

[18] 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. 3

[19] 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. 3

[20] 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. 1

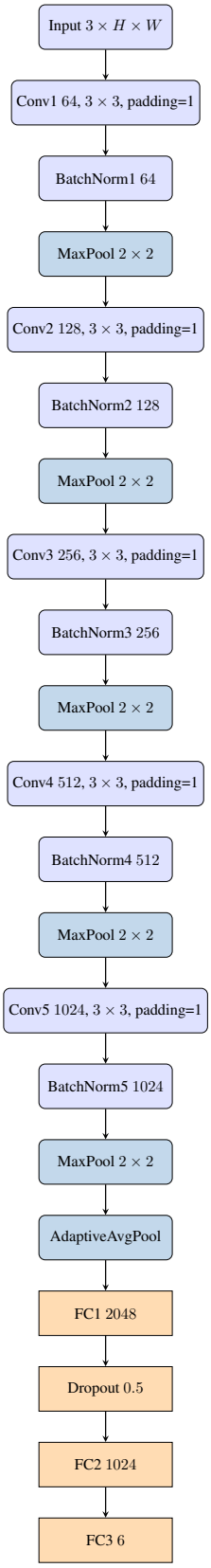

Figure 3. Overview of our detailed model architecture