GiMeFive: Towards Interpretable Facial Emotion Classification

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

Abstract

Deep convolutional neural networks have been shown to successfully recognize facial emotions for the past decade in the realm of computer vision. However, the existing detection approaches are not always reliable or explainable, we propose our model GiMeFive with interpretations, i.e., via layer activations and Grad-CAM. We compare against the state-of-the-art models on five facial emotion benchmarks. Empirical results show that our model outperforms the previous methods in terms of accuracy. Our code and supplementary material are available at `https://github.com/werywjw/SEP-CVDL`.

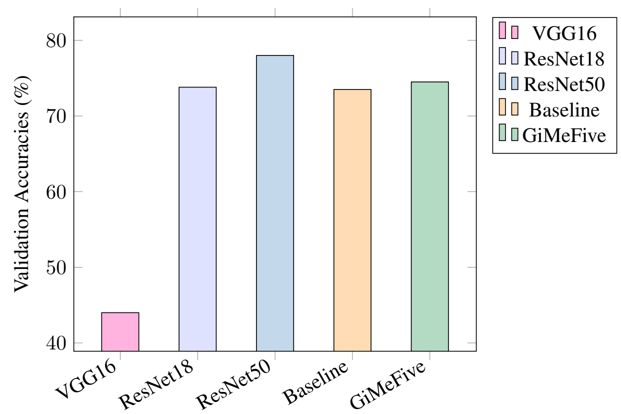

Figure 1. Validation accuracies of our GiMeFive compared to other state-of-the-art models on the aggregated five FER datasets

1. Introduction

Facial emotion recognition (FER) [6, 7, 12, 20] is a topic of significant frontier and ongoing debate, not only in our daily lives but also in the fields of *artificial intelligence* (AI). In this report, we aim to leverage several deep *convolutional neural networks* (CNNs) 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) [17].

Our main contributions can be summarized as follows.

- We collect, preprocess, and evaluate the training and testing data thoroughly, (both image and video) from various public databases.
- We implement all classification models from scratch and optimize them with several techniques in a systematic 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.

Paper Outline. The structure of the 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 hyperparameter tuning. An overview of the experimental pipeline of our project is illustrated in Figure 2. We provide the conclusion and discussion in Section 6.

2. Related Work

Interpretable Emotion Classification. Yin et al. [20] focus on a specific area of interpretable visual recognition by learning from data a structured facial representation. Malik et al. [12]

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) 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 [22] helps interpret CNN decisions by providing visual cues about the

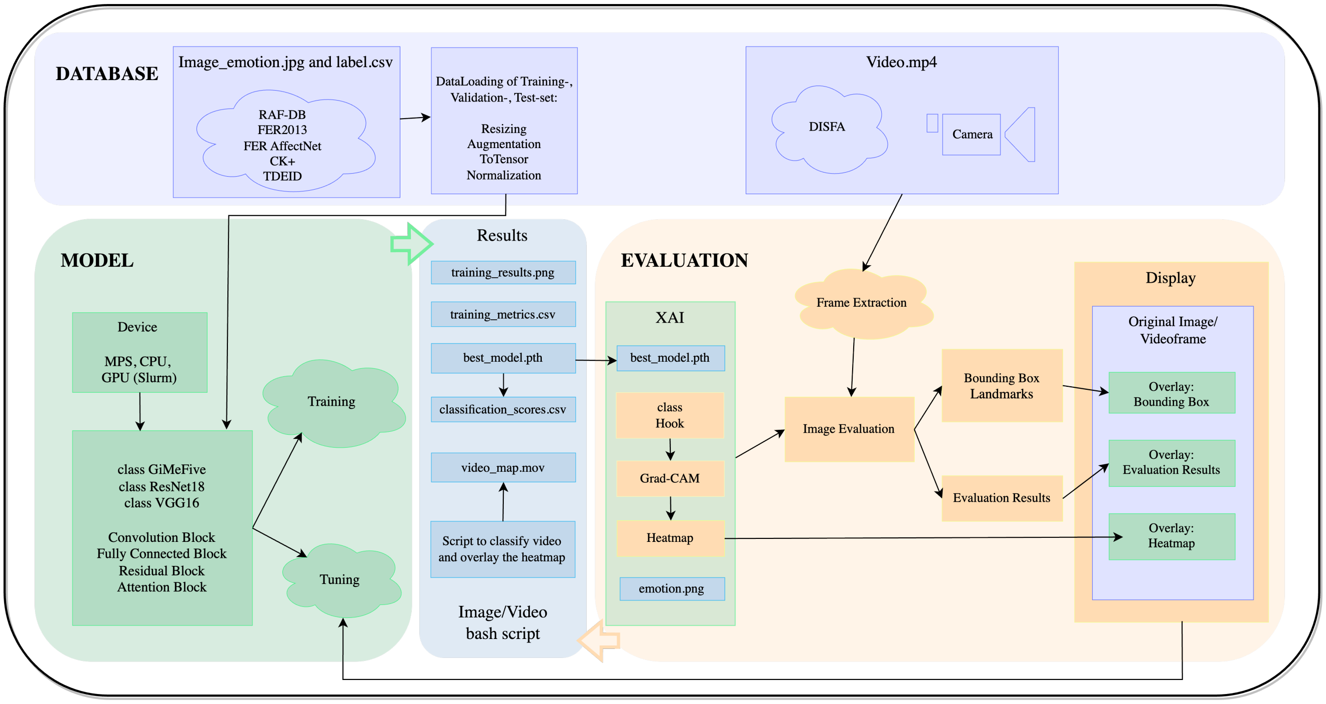

Figure 2. Overview of the experimental pipeline

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. [22] 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++,

3. Experimental Setup

All the experiments are implemented in Python. We also use Shell for generating image and video scripts. The experiment and evaluation are conducted on two MacBook Pro (M1 Pro-Chip with 10-core CPU and 16-core GPU; Intel Core i9 with 2.3 GHz 8-Core).

3.1. Dataset

To initiate the project, we gathered image databases representing different types of emotion expressions. *Real-world Affective Faces Database* (RAF-DB, in-the-wild expression) [8, 9], *Facial Expression Recognition* 2013 (FER2013, real-time wild expression) [1], *FER AffectNet Database* (FER AffectNet, in-the-wild expression) [14], *Extended Cohn-Kanade Dataset Plus* (CK+, posed expressions) [11], and *Taiwanese Facial Expression Image Database* (TFEID, posed expressions) [3, 10]. These image datasets come in folder-structure classification.

For reviewing Explainable AI we used the Video Dataset *Denver Intensity of Spontaneous Facial Action Database* (DISFA, spontaneous expressions) [13], containing a variety of levels of intensity in expression. The procurement was proceeded through public institutions and kaggle [4, 16].

Image Preprocessing. Build upon these image databases, we exclusively analyze human faces representing 6 emotions. That is, we first generalize a folder structure in happiness (0), surprise (1), sadness (2), anger (3), disgust (4), and fear (5). Afterward, we append the emotion labels 0 to 5 to the name of each matching image.

In order to later efficiently pass the images to the model, we also create via script a CSV file to store all the images and their corresponding labels. The images

DATASET	SPLIT	# happiness	# surprise	# sadness	# anger	# disgust	# fear	# total videos/images
DISFA [13]	Test							27
RAF-DB [8, 9]	Train	4772	1290	1982	705	717	281	9747
	Test	1185	329	478	162	160	74	2388
FER2013 [1]	Train	7215	3171	4830	3994	436	4097	23743
	Test	1774	831	1247	958	111	1024	5945
FER AffectNet [14]	Train	3091	4039	5044	3218	2477	3176	21045
CK+ [11]	Train	69	83	28	45	59	25	309
TFEID [3, 10]	Train	40	36	39	34	40	40	229
FER GIMEFIVE	Train	15187	8619	11923	7996	3729	7619	**55073**
	Test	2959	1160	1725	1120	271	1098	**8333**
	Valid	100	100	100	100	100	100	**600**

Table 1. Overview of the data statistics for each emotion class and total number of videos/images in our experiment

together with CSV file are being loaded and preprocessed for training. Therefor we manipulate the pixel data through resizing, augmentation, convertion to grayscale, creating Tensors and normalization. The images are converted to greyscale with three channels at 64×64 resolution in the JPG format, as our original CNN is designed to work with three-channel inputs. Typically, we assume that the color of the image does not affect the emotion classification. For augmentation we determined `RandomHorizontalFlip`, `RandomRotation`, `RandomCrop`, and `RandomErasing`. Our result and analysis is given in Section 5.1.

To enhance the generalizability and robustness of our model, we aggregate the previous five FER benchmarks into a new customed dataset called FER GIMEFIVE (See Table 1 for statistic details with specific numbers for each emotion class). That is, the training set of FER GIMEFIVE is aggregated from the training sets of the five datasets, while the test set is combined from the test sets of RAF-DB and FER2013. The validation set is given with equally 100 images per class to test the performance of models.

Video Preprocessing. To evaluate videos or live webcam streams, we extract every frame and crop the area (rectangle) of interest for emotion detection. The cropped image is preprocessed by resizing to 64×64 resolution, then converted to greyscale with three channels, after that a Tensor is generated, wich then is normalized. For every cropped is evaluated through our model.

3.2. Model Architecture

In order to compare the performance of our model with other state-of-the-art models, we replicate several CNNs from scratch, which includes ResNet18 [5], VGG [18] on the FER GIMEFIVE.

Residual Networks. In principle, deeper neural networks are more difficult to train. Inspired by He et al. [5], we build the ResNet18 with the residual block from scratch to ease the training process.

VGG16. Simonyan and Zisserman [18]

GIMEFIVE. Figure 3 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 the larger kernel being able to provide more information and a 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 share the weights without expensive computation. Following each convolutional layer, batch normalization (BN) is used for stabilizing the learning by normalizing the input to each layer. Meanwhile, BN ensures forward propagated signals to have non-zero variances. 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.

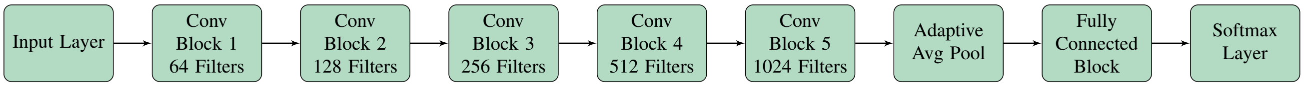

Figure 3. Overview of the GIMEFIVE model architecture (see Figure 4 for a detailed version)

4. Evaluation

For evaluation, we use the metric accuracy to see if our model can classify the facial emotions correctly. We report all the training, testing, and validation accuracies in % to compare the performance of our GIMEFIVE with other state-of-the-art methods.

The loss function employed for all models is cross-entropy (CE), which is typically for multi-class classification. Mathematically, the CE loss is defined as follows:

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

where y_i is the true label and p_i is the predicted probability of the i-th class. Here n denotes the total number of classes, in our case, 6.

4.1. Evaluation Results

As seen in Table 2, the test result are 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.

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, 19, 21].

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.

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.

Heatmap Overlayed GradCAM. 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.

5.2. Hyperparameter Tuning

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.

6. Conclusion and Discussion

7. Limitation

Dataset	Models	# Layers	Architecture	Accuracies			# Parameters
				Train	Test	Valid	
RAF-DB [8, 9]	ResNet18 [5]	18	Residual Block	**98.9**	81.3	67.9	11179590
	VGG16 [18]	16					
	Ours	13	+BN-SE	96.6	80.6	66.8	2606086
	Ours	10	-BN-SE	96.3	76.9	60.6	10474118
	Ours	16	+BN+SE	98.4	81.7	71.1	10478598
	Ours	15	+BN-SE	98.6	**83.1**	**72.1**	10478086
	Ours	17	+BN-SE	97.5	82.5	70.0	41950726
FER2013 [1]	Ours	13	+BN-SE	86.6	64.1	40.2	2606086
	Ours	15	+BN-SE	89.6	**65.6**	40.7	10478086
	Ours	17	+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)

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`, `AdamW`, `SGD`}
Activation	{ `relu`, `tanh`, `elu`, `gelu`}
Epoch	{50, 100}
Early stopping	{`True`, `False`}
Patience	{5, 10, 15}

Table 3. Explored hyperparameter space for our models

Author Contributions

- **Jiawen Wang** implemented different model architectures from scratch, training and evaluation infrastructure, classification score and image labeling script, Grad-CAM explanation, and optimization strategies, together with the corresponding writing part. Also, she is responsible for the figures and tables of this report.
- **Leah Kawka** collected the databases, prepared data processing, provided sever with databases, implemented augmentation and evaluating video/webcam script. She helps run the results with Slurm, (and takes part in the explainable AI)
  In the specific writing part, she also writes the related parts and drew the Figure 2. and took part in Dataset figure
- **Mahdi Mohammadi** implemented the augmentation, did the research searching, conclusion researching, data preprocessing, 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, 5

[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. 2

[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, 4

[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. 3, 5

[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 recogni-

tion 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, 5

[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, 5

[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] Sarthak Malik, Puneet Kumar, and Balasubramanian Raman. Towards interpretable facial emotion recognition. In *ICVGIP '21: Indian Conference on Computer Vision, Graphics and Image Processing, Jodhpur, India, December 19 - 22, 2021*, pages 14:1–14:9. ACM, 2021. 1

[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. 2, 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. 4

[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] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 3, 5

[19] 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. 4

[20] Bangjie Yin, Luan Tran, Haoxiang Li, Xiaohui Shen, and Xiaoming Liu. Towards interpretable face recognition. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 9347–9356. IEEE, 2019. 1

[21] 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. 4

[22] 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, 2

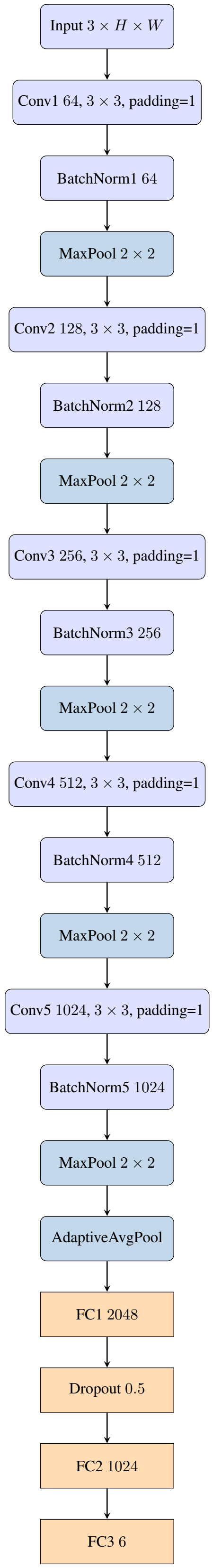

Figure 4. Overview of our detailed model architecture