A Fusion of Self-Cure Network and Transfer Learning-Based Deep Convolutional Neural Network for Facial Emotion Recognition

Tien Phan
Earlham College
Richmond, Indiana, United States
ttphan21@earlham.edu

ABSTRACT

Recent advancements in deep learning, especially the application of Convolutional Neural Networks (CNNs), have guided significant progress in Facial Emotion Recognition (FER). CNNs excel at automatically extracting facial features, leading to the development of robust FER systems. Deep Convolutional Neural Networks (DCNNs) have further elevated FER capabilities, outperforming traditional CNNs. However, DCNNs still face a limitation concerning their demand for high-dimensional datasets. To address this constraint, this paper proposes an ensemble model combining Self-Cure Network (SCN) with DCNN, bolstered by transfer learning.

KEYWORDS

Convolutional neural networks (CNN), deep convolutional neural networks (DCNN), self-cure network (SCN), transfer learning, facial emotion recognition (FER)

1 INTRODUCTION

Facial expressions play a vital role in human communication, acting as a universal language conveying emotions, intentions, and social cues [5]. The accurate recognition and interpretation of these expressions have far-reaching implications across various domains such as psychological treatment or surveillance [9]. Consequently, Facial Expression Recognition (FER) has attracted substantial attention and investment over the years.

Among the methods used in FER, Convolutional Neural Networks (CNNs) have gained immense popularity due to their ability to automatically extract features and adapt to diverse datasets with high precision. Deep Convolutional Neural Networks (DCNNs), a specialized form of CNNs, excel at processing high-dimensional images. However, the widespread adoption of DCNNs has been limited by the substantial computational effort and extensive training data they demand.

To enhance the training process of DCNNs, Akhand et al. [1] introduced a promising approach—pre-training DCNN models through transfer learning. While this method has improved the efficiency and accuracy of DCNNs, it encounters challenges when dealing with datasets containing low-resolution images or highly imbalanced cases. To overcome these obstacles, this paper will propose the fusion of the Self-Cure Network [13] and DCNN models utilizing transfer learning, offering a solution to these limitations.

The paper is structured in four sections: Section 2 provides background information and related works. Section 3 describes the design of the project. Section 4 provides the timeline of the project.

2 RELATED WORK

This section will provide the background knowledge related to the method proposed in this paper including topics like DCNN, transfer learning, and SCN.

2.1 Transfer learning based Deep Convolutional Neural Networks Model

A CNN typically comprises a series of convolutional layers, followed by pooling layers and fully connected layers. DCNNs are a kind of CNN that are characterized by their depth, consisting of multiple layers, including convolutional, pooling, and fully connected layers. These layers collaborate to autonomously learn and extract features from images. Convolutional layers capture basic features like edges and textures, while deeper layers identify complex patterns and structures [1].

Building and training a DCNN from scratch is a heavy task due to its complexity. To overcome this challenge, this project employs the approach proposed by Akhand et al. [1], which involves utilizing pre-trained DCNN models. Some popular DCNN models such as AlexNet [2], VGG, and ResNet [8] have demonstrated exceptional performance in image classification tasks on ImageNet [12], a vast dataset of labeled images. These models are pre-trained on ImageNet, where they acquire the ability to recognize a wide range of objects and features within images. This is where the concept of transfer learning comes into play.

Transfer learning is a machine learning technique that applies knowledge learned from one task to a different yet related task. In DCNN, transfer learning entails taking a pre-trained model, such as one trained on ImageNet, and adapting it to a new task, such as facial emotion recognition. This approach leverages the knowledge acquired during the pre-training phase, providing the model with a strong foundation, even when the new task has limited data available. The method proposed by Akhand et al. [1] underscores the significance of fine-tuning within the context of transfer learning. Fine-tuning is a critical step in transfer learning. It involves modifying the pre-trained DCNN by replacing or adding layers specifically tailored to the new task, such as facial emotion recognition. Typically, the existing convolutional base is retained, and new layers, often fully connected layers, are added to adapt the model to the target task. These newly added layers are initialized with random weights and are then trained using the new dataset.

In the case of transfer learning-based DCNN for FER, the pre-trained model (e.g., VGG-16 trained on ImageNet) receives modifications to suit emotion recognition by redefining its dense layers. Subsequently, fine-tuning is executed using emotion data. In this process, the last dense layers of the pre-trained model are replaced with new dense layers, which are designed to recognize facial images

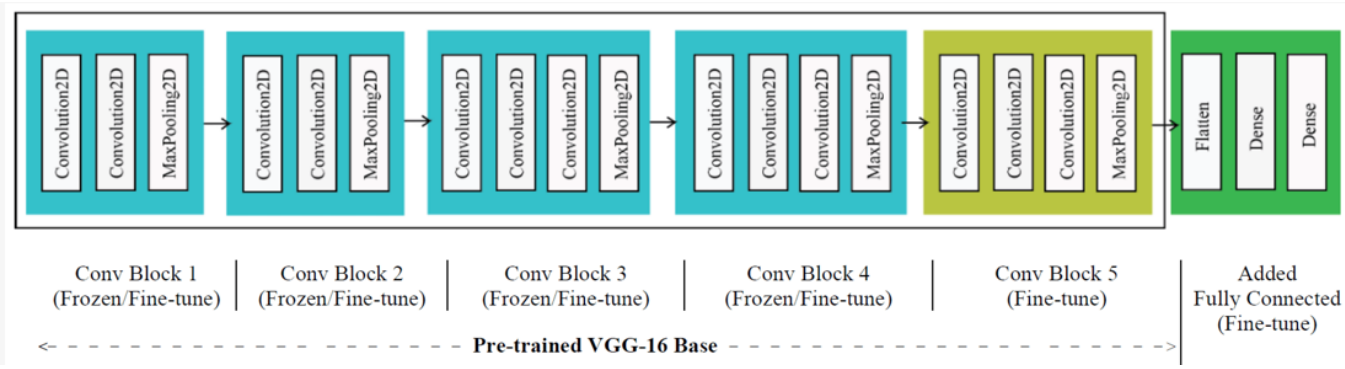

Figure 1: The architecture of DCNN with a detailed pre-trained VGG-16 model with dense layers for FER [1]

categorized into one of seven emotion classes, such as afraid, angry, disgusted, sad, happy, surprised, and neutral.

The complete model, composed of the pre-trained DCNN and the newly added dense layers, facilitates the fine-tuning of dense layers and selected layers of the pre-trained model using emotion data. The pre-trained VGG-16 model, as shown in Figure 1, consists of five convolutional blocks, with each block featuring two or three convolutional layers and a pooling layer.

2.2 Self-Cure Network

Wang et al. [13] introduced the Self-Cure Network (SCN), an extension of conventional CNNs. The primary objective of SCN is to address the challenges arising from the subjective nature of FER datasets obtained from the internet, which often result in inconsistent and erroneous labels. SCN comprises three essential components: self-attention importance weighting, ranking regularization, and relabeling. When given a dataset containing uncertain samples, the system first extracts deep features using a backbone network like CNNs. The self-attention importance weighting module assigns significant weights to each image, while the rank regularization module adjusts the attention weights to downplay the importance of uncertain samples. Lastly, the relabeling module modifies some of the uncertain samples within the low importance group. Figure 2 visually shows the structure of SCN: the process starts by using a backbone CNN to extract features from facial images. Then, a self-attention importance weighting module learns weights for each sample based on facial features, influencing the loss calculation. The rank regularization module uses these weights and applies constraints using a ranking operation and a margin-based loss function. In the labeling module, reliable samples are identified by comparing predicted probabilities with given labels. Mislabeled samples are marked with red rectangles, and ambiguous ones with green dashed rectangles. It's important to note that the Self-Cure Network mainly uses re-weighting to handle uncertainties and adjusts only certain samples.

To assess the effectiveness of Self Cure Network (SCN), the study conducted experiments using four distinct datasets: RAF-DB [10], FERPlus [3], AffectNet [11], and a dataset referred to as WebEmotion, which contained data collected from the Internet. The primary focus was on addressing the challenge of incorrect or noisy annotations, which led to the creation of the WebEmotion dataset. It is a video dataset that was originally obtained from YouTube, but for the purpose of this study, it was treated as an image dataset by assigning emotion labels to individual frames. The dataset was curated by searching for videos using a set of 40 emotion-related keywords, combining them with 45 country-related keywords including countries in Asia, Europe, Africa, and America and six age-related keywords (baby, lady, woman, man, old man, old woman). The WebEmotion dataset comprises the same eight emotion classes as FERPlus [3], with each class being associated with various emotion-related keywords. For example, the "happy" class is linked to keywords such as happy, funny, ecstatic, smug, and kawaii. To establish meaningful correlations between these keywords and the collected videos, only the top 20 retrieved videos, each lasting less than four minutes, were selected for inclusion in the dataset. This curation process resulted in approximately 41,000 videos, which were further segmented into 200,000 video clips, with the requirement that a human face, detected using the Multi-task Cascaded Convolutional Neural Networks (MTCNN) [14] face detection method, must appear for at least five seconds in each clip. Moreover, they employed ResNet-18 [8] as the foundational network, which had been pre-trained on the MS-Celeb-1M face recognition dataset [7]. The extraction of facial features was performed from the final pooling layer of this network.

3 DESIGN

In this project, an ensemble approach will be implemented, combining the SCN with a transfer learning-based DCNN using a stacking strategy. The results obtained from this ensemble method will be compared with the existing state-of-the-art methods to assess its performance.

3.1 Implementation

For the implementation of this project, the integration of the Self-Cure Network (SCN) and the Deep Convolutional Neural Network (DCNN) will be achieved through a defined process. Both the SCN and DCNN will share the same pre-trained model, ResNet-18, which has previously been trained on the extensive MS-Celeb-1M dataset. To ensure data compatibility, image resizing will be done using

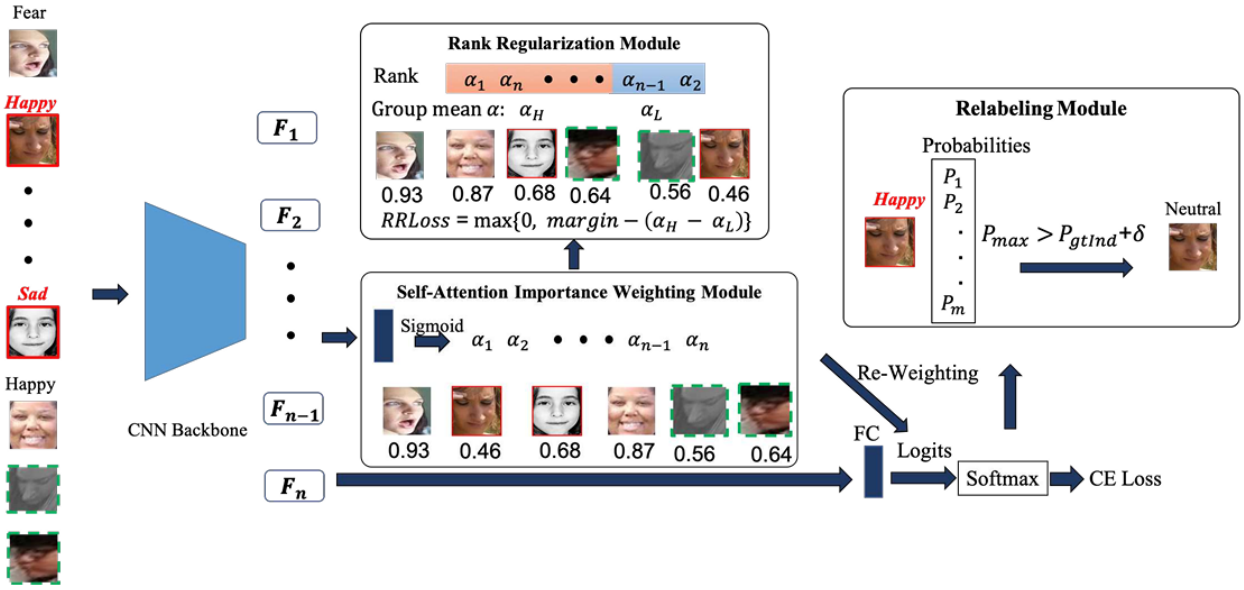

Figure 2: The architecture of SCN [9]
The images are fed into a CNN backbone for feature extraction. The self-attention weighting module assigns weights to these features. The rank regularization module takes the weights and constraint them with ranking operation and a margin-based loss function. The relabeling module compare maximum predicted probabilities to the probabilities of given labels to find reliable samples. Mislabeled samples are indicated with red line, and ambiguous ones with green dashed line.

PyTorch. Both models will also have the setting with their optimizer being stochastic gradient descent or SGD and the learning rate equal to 0.001. DCNN has 24 epochs and SCN has 70 epochs. The resized images will serve as inputs for the DCNN, which features the ResNet-18 model along with the addition of new dense layers for FER-2013 and KDEF. The output of DCNN will be processed only into the training phase of SCN, but not the testing phase. The next step involves channeling the DCNN's output into the SCN, which will also make use of the same pre-trained ResNet-18 model. The SCN's role is to address uncertainties and improve the overall quality of predictions, so the project will utilize the modules of SCN, which happens during the training process. Hence, the output of DCNN will be loaded into the training phase with both the predicted labels from DCNN and the target labels. The last module will work on relabeling on the predicted label array from DCNN based on prediction made from ResNet18. At the end, the result generated by the SCN will serve as the final outcome for comparison and evaluation against the original models.

3.2 Datasets

The project will use KDEF [4] dataset, which was developed by Karolinska Institute, Department of Clinical Neuroscience, Section of Psychology, Stockholm, Sweden. The dataset contains 4408 training images and 490 testing images with high-resolution and clear labels of standard facial expression from multiple angles. KDEF includes 7 emotions: AF (afraid), AN (angry), DI (disgust), HA (happy), NE (neutral), SA (sad), SU (surprise).
In addition to that, the project will use the FER-2013 [6] dataset, which was derived from the Google search engine and consists of a large set of faces automatically registered and labeled with basic emotions. The images are more varied in terms of quality, angle, lighting, and occlusion. FER-2013 includes 7 emotions: angry, disgust, fear, happy, neutral, sad, surprise. This dataset has 28709 training images and 7178 testing images.

3.3 Project Setup

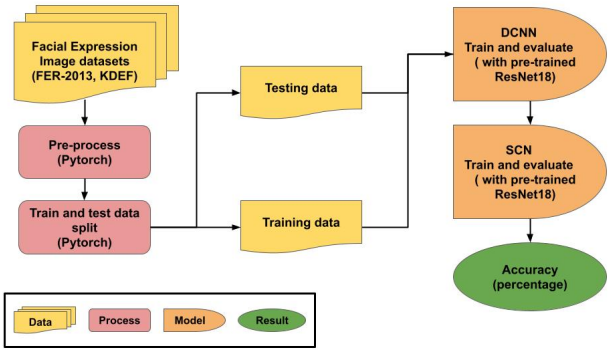

Figure 3: Data Architecture Diagram

Before training the models, I followed the data transformation based on DCNN code source [1]. For training data, the images are moderated using `torchvision` model from PyTorch. First, it is resized to 224x224 pixels since this is a common input size for many CNNs. After that, images are randomly flipped horizontally

and converted to a torch.`FloatTensor` of shape (CxHxW). Images are also normalized with a mean and a standard deviation: the normalization values for the mean are `[0.485, 0.456, 0.406]` and for the standard deviation are `[0.229, 0.224, 0.225]`. These values are specific for model that has been pretrained on ImageNet dataset, which is ResNet18 for this project.
For testing data, images are resized to `256x256` pixels and cropped at the center to `224x224` pixels. After that, testing images follow the same conversion and normalization conditions as training images. After preparing the data, the next step is loading the data using `DataLoader` from `PyTorch`, which specify that each batch of data will contain 4 images and after each epoch, the images will be shuffle to prevent the model from the learning the order of the images.

4 RESULT

4.1 DCNN

After training and evaluating on DCNN, the best accuracy for FER-2013 is 65%:

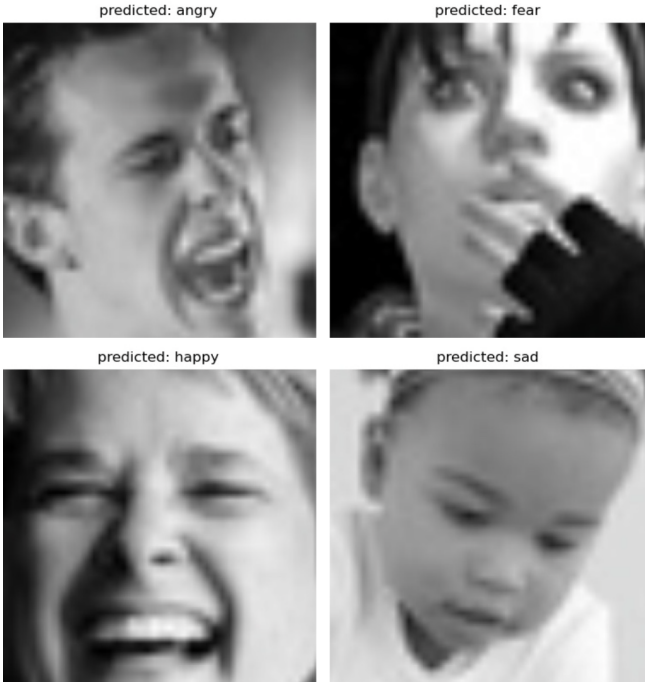

Figure 4: FER-2013 visualization

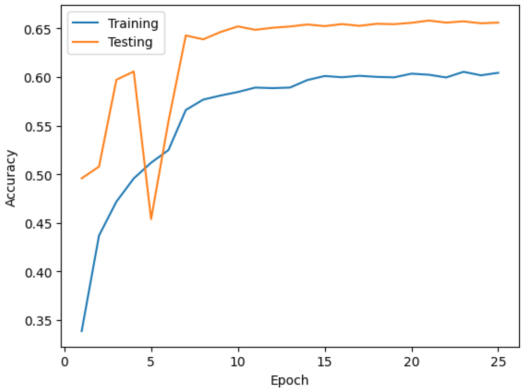

Figure 5: DCNN on FER-2013 model accuracy

The best accuracy for KDEF if 91%:

Figure 6: KDEF visualization

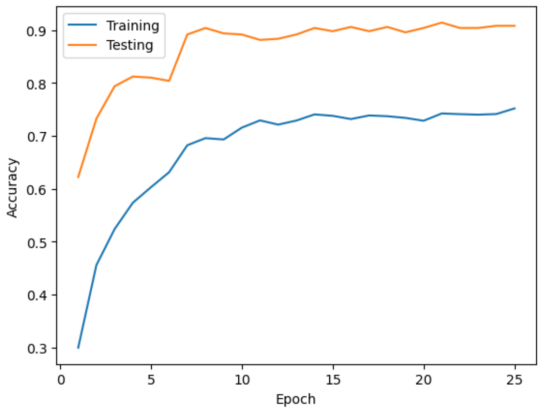

Figure 7: DCNN on KDEF model accuracy

4.2 SCN

When analyzing the performance of the original model on the FER-2013 and KDEF datasets, I notice that the model achieves 97% training accuracy on FER-2013 but only 68% on testing, while it scores around 96% on KDEF for both training and validation. This gap in performance on FER-2013 can be attributed to the use of a relabeling module during training. This module adjusts the training labels based on predictions from a pre-trained model, aiming to correct mislabeled data. However, while this helps in enhancing the reliability of the training data, it doesn't necessarily improve the model's ability to generalize to new, unseen data. This suggests that the high training accuracy may be due to the model overfitting to the corrected labels rather than achieving true predictive improvements, thus explaining the lower test accuracy. The better performance of the model on the KDEF dataset as compared to FER-2013 could also be influenced by the size of these datasets. KDEF is significantly smaller than FER-2013, which might make it easier for the model to fit well to the limited variety of examples available in KDEF, resulting in higher accuracy.

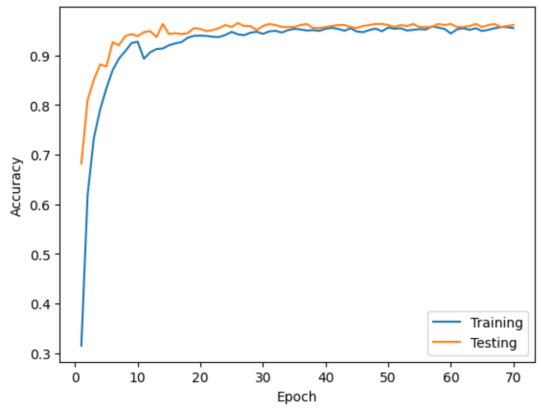

Figure 8: SCN on KDEF model accuracy

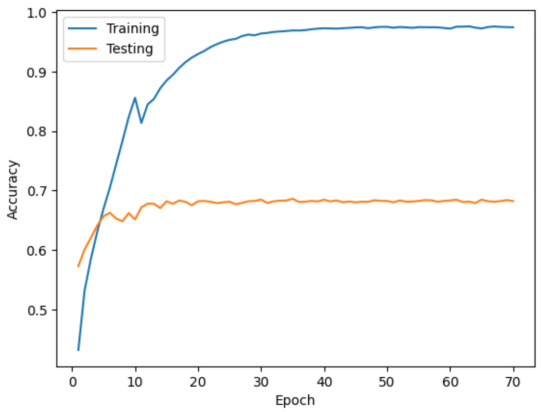

Figure 9: SCN on FER-2013 model accuracy

4.3 SCN intergrating with DCNN

In the method proposed, both the predicted labels from DCNN and the actual target labels on the testing dataset are input into the training phase of SCN. This approach is designed to leverage the relabeling module offered by SCN. However, as discussed previously, this relabeling module doesn't directly train the model to improve its prediction capabilities; instead, it adjusts the labels based on the predictions from the DCNN, aiming to enhance label accuracy during training.

Because the relabeling function of the SCN modifies training labels manually rather than improving the model's ability to generalize, I opted to include only the output from the DCNN in the SCN's training phase and excluded it from the testing phase. This decision is based on the understanding that while the relabeling can improve performance by cleaning up the labels, it does not necessarily equip the model with the skills needed to perform better on test data.

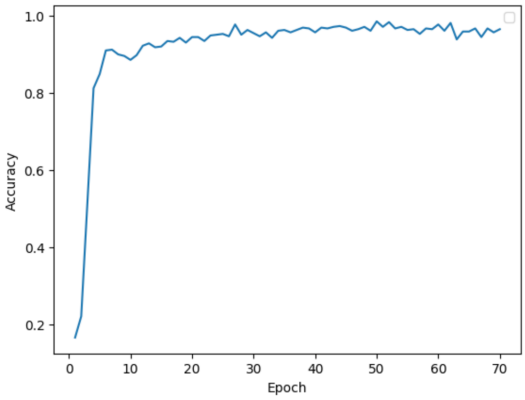

Figure 10: SCN combined DCNN on KDEF model accuracy

In this process, the testing datasets from KDEF and FER-2013 are utilized during the training phase of SCN. Despite KDEF having 490 samples and FER-2013 having 7178 samples, the accuracy graphs

exhibit similar trends to the original model. There's a slight deviation in the curve for FER-2013, but both datasets ultimately achieve high accuracy, with KDEF reaching 98% and FER-2013 hitting 97%.

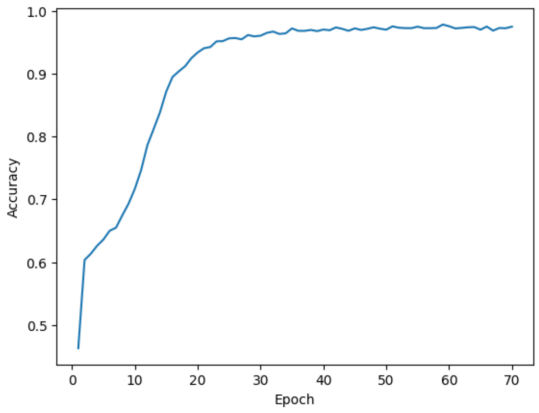

Figure 11: SCN combined DCNN on FER-2013 model accuracy

Overall, while using SCN on the output from DCNN does enhance accuracy, there might be a bias in the process. This is because the improvements depend more on the modules in SCN's training phase rather than the capabilities of the SCN model itself.

5 FUTURE WORK

The project encounters challenges due to limited expertise in integrating deep learning models, potentially leading to oversights during integration and evaluation. Therefore, improving the methods used to combine these models is essential for progressing in this area. Moreover, it is crucial to test this approach on larger datasets to verify its effectiveness. Although the KDEF dataset has shown high accuracy with training using DCNN, the scope for further improvement might be limited when integrating with SCN. Nevertheless, the SCN's modules show promise in enhancing predictions after receiving outputs from the DCNN. If these modules can be effectively integrated into the testing phase, where more data refinement is needed, and can further optimize the training phase by adjusting the pre-trained model, this could be a beneficial strategy to explore and test.

ACKNOWLEDGMENTS

I would like to thank Dr. David Barbella and Dr. Charlie Peck for their supports and feedbacks on this project.

REFERENCES

[1] M.A.H. Akhand, Shuvendu Roy, Nazmul Siddique, Md Abdus Samad Kamal, and Tetsuya Shimamura. 2021. Facial emotion recognition using transfer learning in the deep CNN. *Electronics* 10, 9 (2021), 1036.

[2] Grigorios Antonellis, Andreas G. Gavras, Marios Panagiotou, Bruce L Kutter, Gabriele Guerrini, Andrew C. Sander, and Patrick J Fox. 2015. Shake table test of large-scale bridge columns supported on rocking shallow foundations. *Journal of Geotechnical and Geoenvironmental Engineering* 141, 5 (2015), 04015009.

[3] Emad Barsoum, Cha Zhang, Cristian Canton Ferrer, and Zhengyou Zhang. 2016. Training deep networks for facial expression recognition with crowd-sourced label distribution. In *Proceedings of the 18th ACM international conference on multimodal interaction*. 279–283.

[4] Manuel G Calvo and Daniel Lundqvist. 2008. Facial expressions of emotion (KDEF): Identification under different display-duration conditions. *Behavior research methods* 40, 1 (2008), 109–115.

[5] Chris Frith. 2009. Role of facial expressions in social interactions. *Philosophical Transactions of the Royal Society B: Biological Sciences* 364, 1535 (2009), 3453–3458.

[6] Ian J Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. 2013. Challenges in representation learning: A report on three machine learning contests. In *Neural Information Processing: 20th International Conference, ICONIP 2013, Daegu, Korea, November 3-7, 2013. Proceedings, Part III 20*. Springer, 117–124.

[7] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. 2016. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*. Springer, 87–102.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[9] Yunxin Huang, Fei Chen, Shaohe Lv, and Xiaodong Wang. 2019. Facial expression recognition: A survey. *Symmetry* 11, 10 (2019), 1189.

[10] Shan Li, Weihong Deng, and JunPing Du. 2017. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2852–2861.

[11] Ali Mollahosseini, Behzad Hasani, and Mohammad H. Mahoor. 2017. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing* 10, 1 (2017), 18–31.

[12] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[13] Kai Wang, Xiaojiang Peng, Jianfei Yang, Shijian Lu, and Yu Qiao. 2020. Suppressing uncertainties for large-scale facial expression recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6897–6906.

[14] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters* 23, 10 (2016), 1499–1503.