Proceedings of the 2023 International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/8/20230296

Facial expression recognition model based on CNN and data augmentation method

Zhewei Deng

College of Civil Engineering, Fuzhou University, Fuzhou, Fujian Province, 350108, China

052105127@fzu.edu.cn

Abstract. Face expressions are essential in expressing human emotions, and it is accomplished by separating features and categorizing them. Facial expression recognition technology has been widely employed in human-computer interaction, telemedicine, mental health, and criminal investigation detection. In recent years, significant advances in deep learning have facilitated the development of facial expression recognition, making it increasingly accessible. Convolutional neural networks (CNN) are the foundation of the face expression recognition model presented in this article. In order to maximize the final accuracy, an abundance of sample photographs are required for training and optimization. As a result, the 35,886 facial expression photos from the FER2013 dataset, which contains all seven emotions, were used for both training and testing. The photos were scaled down to 48×48 pixels during data preparation, and data augmentation was carried out. To get accurate face expression recognition results, various optimizers and parameters were chosen for the training network, which was a bespoke structure based on the VGG network design. The model constructed in this study achieved an accuracy of 73.16% during prediction.

Keywords: facial expression recognition, CNN, confusion matrix, data augmentation.

1. Introduction

Professor of psychology Albert Mehrabian argued that there are six basic human emotions, each of which is associated with a different psychological activity: anger, happiness, sorrow, surprise, disgust, and fear. Subsequently, a neutral feeling was included as the seventh. Humans can express their feelings through language, tone, and facial expressions, which is the fundamental reason why they are able to communicate emotions. More information is communicated through facial expressions than through language and tone combined, 55% of the time [1]. A deeper comprehension of and ability to use human facial expressions is made possible by studying facial expression recognition. The use of facial expression recognition technology can enable better human-computer interaction, such as reducing distance in caring for the elderly and young children. This technology can also be applied to emotional robots, monitoring of critically ill patients, online education, and other areas, making computers better serve humanity [2].

With a straightforward convolutional neural network (CNN) approach, this study aims to develop a computer algorithm that can recognize facial expressions in people. The VGGnet architecture serves as the foundation for the main CNN model, which is then altered to fit the dataset's properties. The CNN

DOI: 10.54254/2755-2721/8/20230296

model achieves a standard of optimization by parameter and structure optimization based on experimental findings, and its image recognition capability has strong model generalization ability, straightforward experimental operation, and high accuracy [2], making it particularly ideal for the task of recognizing face expressions. In this paper, the CNN model is trained and optimized using the FER2013 dataset, which enables the precise recognition of facial emotions in photographs.

From the perspective of the FER2013 dataset, the data imbalance in FER2013 may lead to lower accuracy when distinguishing emotions with fewer data samples, while higher accuracy may be achieved when analyzing emotions with more original data or data augmented samples.

Agrawal et al. used bespoke VGGnet trained on FER2013 has a 65% accuracy rate [3]. To remove imbalance-induced bias, Jiang et al. employed the softmax loss approach in addition to the ASL and obtained an accuracy of 70.80% on the FER2013 [4]. In order to successfully avoid the negative effects from characteristics of several datasets during model training, Wang et al. used multiple face recognition datasets, including FER2013, JAFFE, RAF-DB, and SFEW [5]. They also substituted the fully-connected layer with CRBM, which allowed them to achieve an accuracy of 73.75% on FER2013. Qiao et al. combined CNN and SVM for training on the FER2013 and CK+ facial expression datasets, achieving an accuracy of 73.4% on the FER2013 dataset [6].

Based on other experimental data and model analysis, I estimate that the model can also achieve an accuracy of around 70% after data augmentation. Moreover, the accuracy of 'happy' and 'disgust' is expected to be higher than that of other emotions.

2. Method

This project uses PyCharm to provide a Torch platform for testing in the Windows 11 environment, allowing for the achievement of seven different types of expression recognition.

2.1. Data and preprocessing

The FER2013 dataset, which is frequently used for facial expression recognition, was generated from the Kaggle facial expression recognition competition in 2013. Each image in the dataset has been annotated, and seven expressions have been labeled in accordance with the expressions in the image.

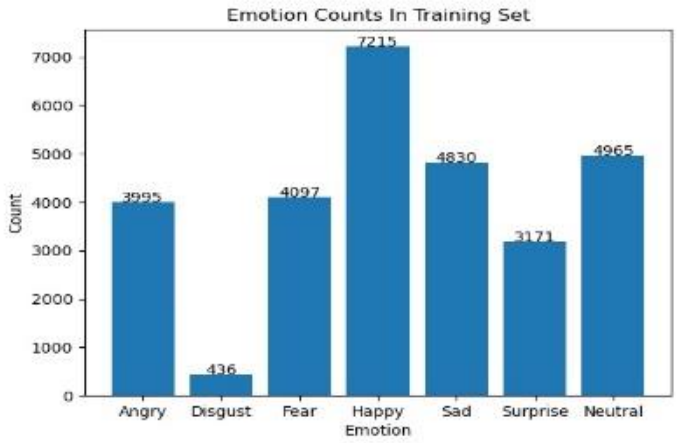

Figure 1. The data counts in the original dataset.

As shown in Figure 1, the facial expression data in FER2013 is highly imbalanced, disgust having significantly fewer data compared to other emotions, with only 436 images. During model training, the features of these rare classes of data are frequently repressed by the features of frequent classes, which can significantly affect the training process [7]. After the preprocessing steps, the quality and quantity of input data are expected to improve, redundancies are reduced or even eliminated. To avoid the neural network from becoming too thick, photos in the FER2013 dataset are first transformed into 48×48 grayscale images before being input into the network. In real-life situations, collected image data may vary in terms of direction, position, size, visibility, and so on. Standardizing and cropping these images

DOI: 10.54254/2755-2721/8/20230296

using common preprocessing techniques can enhance their recognition in experiments [8].

In this study, the data preprocessing involved center cropping and corner cropping of images in the dataset, followed by conversion to tensor format. Standardization was then performed on each image using mean and variance to scale the tensor values. Random cropping, translation, rotation, scaling, and erasing were also applied to generate an improved disgust class for training purposes. The enhanced dataset is shown in Figure 2.

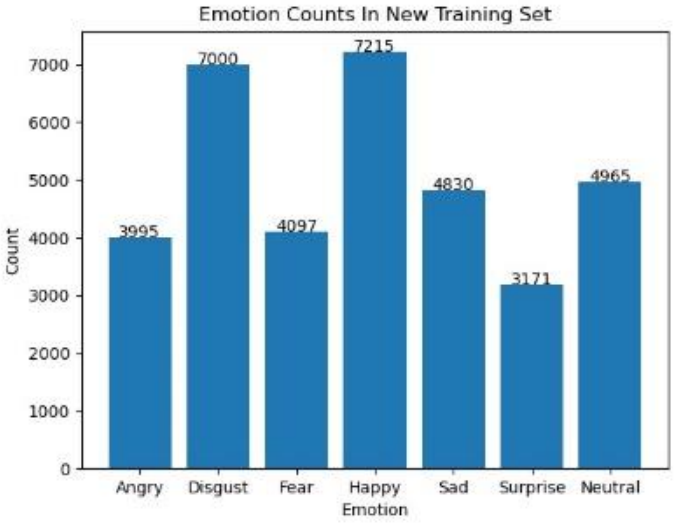

Figure 2. The data counts in the enhanced dataset.

2.2. Convolutional Neural Network (CNN)

With various uses in areas including computer vision and natural language processing, CNN—a feedforward neural network that is classified as the basic operation of convolution—is one of the representative deep learning algorithms. For a while now, the dominant technique for recognizing images has been the Convolutional Neural Network (CNN), a type of deep learning network that incorporates multiple layers and draws inspiration from the way the human brain is structured, as researchers have studied it. By varying the weight coefficients that connect them, neural networks, which are made of several interconnected neurons, can strengthen or reduce models between various neural networks [8,9].

As it presented in Figure 3, the fundamental structure of a convolutional neural network (CNN) comprises three distinct types of layers: convolutional layers, pooling layers, and fully-connected layers, in addition to the input and output layers.

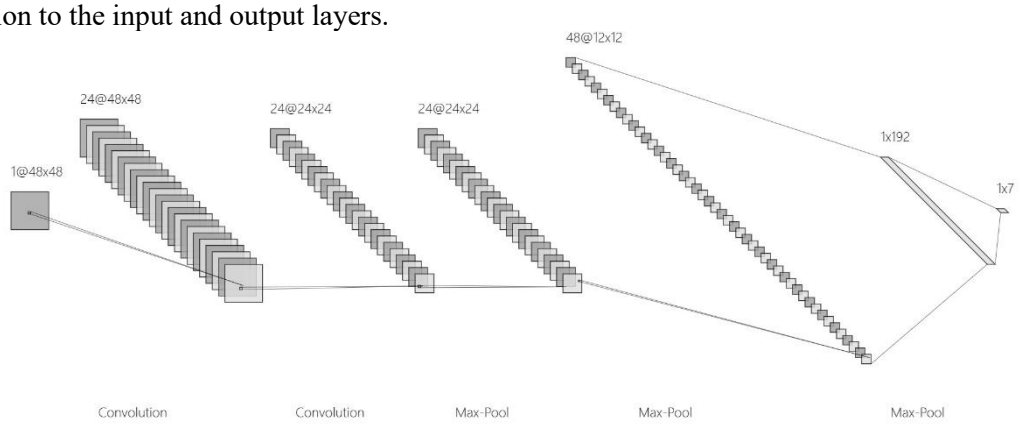

Figure 3. Standard Convolutional Neural Network.

DOI: 10.54254/2755-2721/8/20230296

2.2.1. Convolutional layer. The convolutional layer is a fundamental constituent in deep neural networks that is devised to learn and extract features from input signals. To create a feature map, this layer convolved a group of adjustable filters with the incoming data. Each neuron within a feature map corresponds to a particular small area of the input image, and its activation is determined by applying a nonlinear function to the weighted sum of activations from the previous layer that fall within its receptive field. The convolutional layer employs a sliding-window operation over the input data, where each filter is applied to every feasible location of the input. At each location, the element-wise product of the filter and the corresponding input region is computed, and the resulting values are aggregated to produce a single output element. This operation is repeated for all possible locations of the filter, and the resulting output elements are arranged in a novel feature map. The learnable weights of the filters are updated during the training phase to optimize the performance of the network in the given task. The number of filters is typically determined by the number of distinct features to be extracted, and the strength and diversity of the filters are critical factors that determine the learning capacity and accuracy of the network. As the filter weights are shared across all input locations, there are a lot fewer parameters to learn, which boosts the network's capacity to generalize [9,10].

2.2.2. Pooling layer. A pooling layer often makes up the second layer in a convolutional neural network. In a convolutional neural network, the pooling layer is usually placed between convolutional layers, and it randomly accesses a set of non-overlapping rectangles in the spatial domain between convolutional layers using a non-linear method. The parameter matrix's size, their dimensionality, the number of output feature maps and possibly even the number of parameters in the fully-connected layer are all drastically reduced as a result. Moreover, pooling layers are crucial for accelerating computation and avoiding overfitting. The most popular pooling method used in this experiment is max pooling, while other pooling methods include averaging pooling and summation pooling [9-11]. Max-pooling involves taking the maximum value over a small rectangular neighborhood, typically with a window size of 2x2, applied non-overlappingly to the consequence of a preceding convolutional layer to achieve down sampling. The resulting outcome represents the maximum activation within each pooling window. Figure 4 is an example of Max-Pooling.

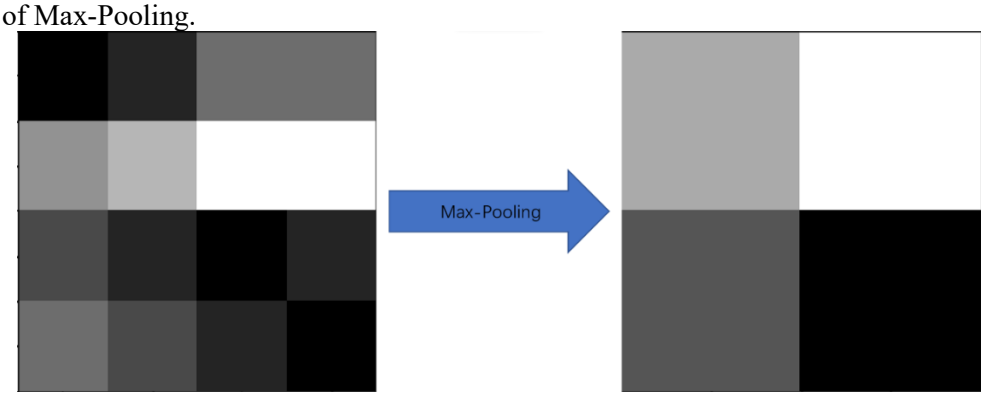

Figure 4. The function of Max-Pooling.

2.2.3. Fully-connected layer. The fully-connected layer's function, in contrast to the preceding convolutional and pooling layers, is to categorize an image's feature matrix rather than extract features from it. Following convolutional and pooling layers, the previous layers are combined to create a fully-connected layer. The fully-connected layer does not store spatial information since every neuron in the current layer is linked to every neuron in the two layers before it. The fully-connected layer combines local knowledge from the preceding layers, with class discrimination, to synthesize the features derived from them. A highly refined set of features is then produced by mapping the retrieved features to the sample label space using the fully-connected layer [8,9]. If it's challenging to analyze a CNN, transforming the convolutional layers into fully-connected layers and examining the behavior of the CNN in a fully-connected manner may facilitate a better understanding of its behavior [6].

DOI: 10.54254/2755-2721/8/20230296

2.3. Evaluation matrix

2.3.1. Confusion matrix. The goal of this article is to assess the effectiveness of a CNN model by utilizing a confusion matrix. A confusion matrix is a tabular representation that is utilized for assessing the performance of a classification model by contrasting its predicted output with the actual or true output. The confusion matrix facilitates the comprehension of the model's efficacy and aids in pinpointing areas that require enhancement. It allows for an evaluation of the model's performance on each category separately, which is very useful when working with datasets that are imbalanced and have different numbers of observations in each category.

3. Result

3.1. Result of the model

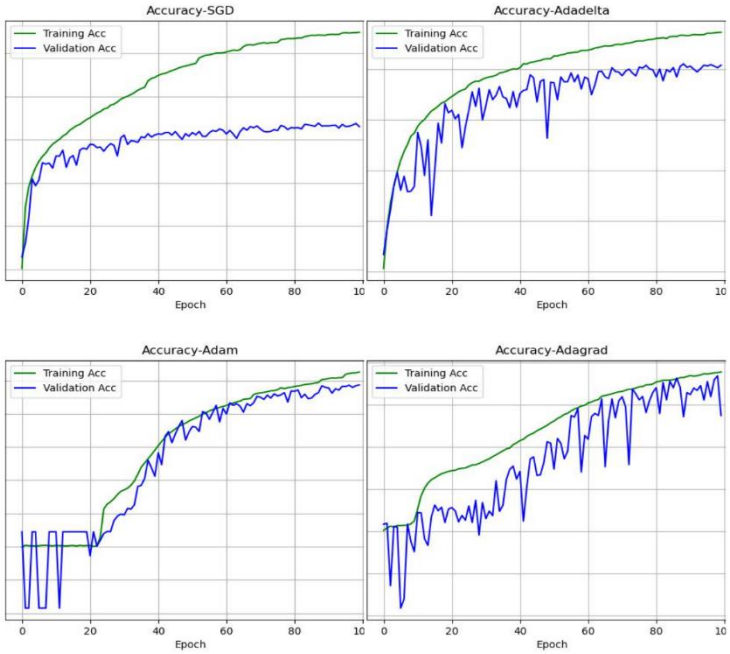

Figure 5. Training process curves on FER2013.

This study employs four optimizers, namely 'Adadelta', 'Adagrad', 'Adam', and 'SGD', to guide the loss function in updating the parameters in the correct direction and appropriate magnitude to gradually converge towards the global minimum. The accuracy changes for each model are displayed in the Figure 5, where each model has undergone 100 epochs of training. The blue curve shows recognition accuracy on the validation set, while the green curve represents recognition accuracy on the training set.

Among these four optimizers, 'SGD' achieves the highest accuracy of 99.54% on the training set and 73.16% on the validation set. The accuracy of both sets increases rapidly in the initial 5 epochs, and the accuracy on the training set continues to rise until it reaches almost 100%, while the accuracy on the validation set increases slowly and stabilizes at around 73% after reaching 70% at around 30 epochs. The reason for this phenomenon may be that the SGD optimizer progresses slowly in the flat dimensions, and when it encounters a local optimal point or a saddle point, the gradient becomes zero, and the parameters cannot be further updated.

The Adadelta optimizer demonstrates the second-highest level of performance, with a training accuracy of 81.26% and a validation accuracy of 70.53%. Both accuracies increase slowly in the first 40 epochs, and the validation accuracy stops increasing after reaching 70%, with some overfitting. This may be due to Adadelta being too sensitive to noise and details in the training set.

DOI: 10.54254/2755-2721/8/20230296

The Adam optimizer performs the third best, with a slow increase in accuracy in the first 20 epochs due to the adaptive learning rate algorithm making it difficult to correct the overfitting results in the early stage. However, the accuracy gradually stabilizes and then quickly rises to around 55%, with the validation and test accuracies steadily increasing to 68.79% and 73.12%, respectively. The low accuracy may be due to the Adam optimizer having a low learning rate in the later stage and requiring more time for further training.

The Adagrad optimizer exhibited the poorest performance, showing severe overfitting during the training process, and achieving the lowest accuracy on both the training and validation sets, with only 57.78% and 48.32%, respectively. Although it demonstrated a relatively fast increase in accuracy, its performance was highly unstable, likely due to the gradual reduction in learning rate and slow updates to the gradients.

Table 1. The Confusion Matrix of the test result.

Labels	Anger	Disgust	Fear	Happy	Sad	Surprise	Normal
Anger	0.67	0.03	0.09	0.02	0.11	0.01	0.05
Disgust	0.02	0.97	0.01	0	0	0	0
Fear	0.13	0	0.63	0.02	0.13	0.06	0.06
Happy	0.02	0	0.01	0.89	0.03	0.03	0.03
Sad	0.08	0	0.12	0.02	0.58	0.01	0.13
Surprise	0.02	0	0.07	0.02	0.01	0.88	0.01
Normal	0.04	0	0.07	0.03	0.12	0.01	0.72

Table 1 presents the confusion matrix of the recognition results of the model on the FER2013 dataset. As shown in Figure 3, disgust, happy, and surprise can be classified with relatively high accuracy, with accuracies of 97%, 89%, and 88%, respectively. However, fear, anger, and sad are relatively difficult to recognize, with success rates of only about 63%, 6Figure7%, and 58%, respectively. Specifically, anger has a 13% chance of being misclassified as fear, fear has a 12% chance of being misclassified as sad, and sad has an 11% chance of being misclassified as anger, a 13% chance of being misclassified as fear, and a 14% chance of being misclassified as normal. Normal recognition performance among the seven expressions is moderate, with an accuracy of 72%, and there is a 13% chance of being recognized as sad.

3.2. Compare the result to the assumption.

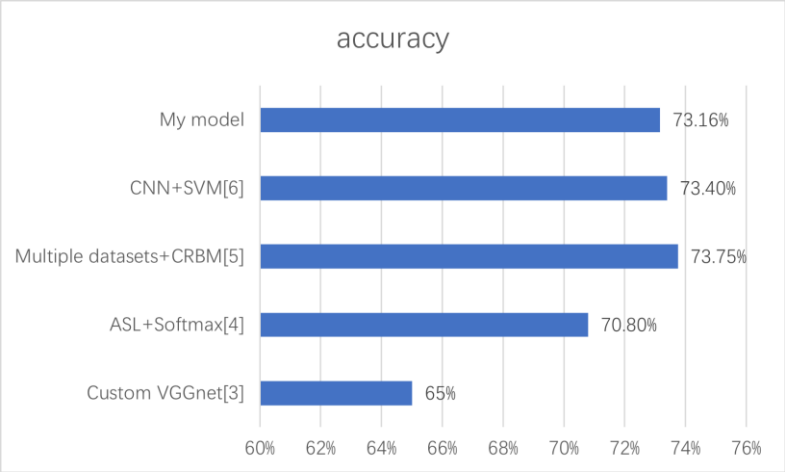

Figure 6. Comparison with other articles results.

DOI: 10.54254/2755-2721/8/20230296

The results of this experiment are consistent with the initial hypothesis. The accuracy of the model in this study was 73.16%, which is higher than the expected 70%. The model achieved high accuracy not only in predicting 'happy' and 'disgust', but also in predicting 'surprise'. Moreover, during the training of the model, the accuracy even reached close to 100%.

As you can see in Figure 6, in terms of accuracy recognition, the model proposed in this paper shows a decrease of 0.24% compared to the results reported in reference [6], a decrease of 0.59% compared to reference [5], an increase of 8.16% compared to reference [3], and an increase of 2.36% compared to reference [4].

Notably, compared to the highest accuracy achieved in reference [12], the model only exhibits a slight decrease of 0.1% in predicting 'anger' and 'fear', while achieving a significant improvement of 21% in predicting 'disgust' and a slight improvement of 0.08% in predicting 'surprise'. These results demonstrate the practical usability of the model in facial expression recognition.

4. Discussion

When compared to other methods, it can be seen that the strategy and model used in this study display relatively good accuracy. This confirms the crucial role of data augmentation in enhancing model accuracy, while also indirectly highlighting the strong performance of CNN models in image recognition tasks.

As the current approach only uses a single CNN model and data augmentation, incorporating additional methods such as SVM or SDM as classifiers may potentially increase the experimental accuracy. Moreover, FER2013 dataset contains some irrelevant erroneous images, and conducting data cleaning at the outset could greatly aid in enhancing the accuracy.

5. Conclusion

The research paper introduces a technique for identifying facial expressions that relies on convolutional neural networks (CNN) and data augmentation methods. The proposed method employs convolutional layers, pooling layers, and fully-connected layers to establish a facial recognition system based on CNN.

The study selects the Max-Pooling method in the pooling layer. After comparing the performance of four different optimizers in terms of accuracy, the method with the highest accuracy, namely, SGD, is employed. The study uses the FER2013 dataset for training, achieving an accuracy of 73.16%, which is among the highest accuracies reported in other papers that use the FER2013 dataset and CNN-based methods. A confusion matrix is employed to analyze the results. The suggested technique successfully recognizes facial expressions with accuracy. The research sheds light on the development and execution of a facial recognition system that employs CNNs for identifying facial expressions.

The limitations of this study are also apparent. Due to the low resolution of the photos in the FER2013 dataset, the accuracy of recognizing some expressions is too low, and the overall identification rate is not optimal. In future experiments, more methods should be attempted to enhance and clean the data, and more advanced methods such as SVM should be employed, instead of relying solely on CNN to produce results. In addition, the results presentation does not include predictions of real-world images, which is an area for improvement.

References

[1] MEHRABIAN A. Communication without words [J]. *Psychology Today*, 1968,2(4): 53—56

[2] Yu Rui. Analysis of facial expression features based on depth learning[D]. College of Computer of Chongqing University, 2018

[3] Agrawal A, Mittal N. Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy[J]. *The Visual Computer*, 2020, 36(2): 405-412.

[4] Jiang P, Liu G, Wang Q, et al. Accurate and reliable facial expression recognition using advanced softmax loss with fixed weights[J]. *IEEE Signal Processing Letters*, 2020, 27: 725-729.

[5] Wang Y, Li Y, Song Y, et al. The application of a hybrid transfer algorithm based on a convolutional neural network model and an improved convolution restricted Boltzmann

DOI: 10.54254/2755-2721/8/20230296

machine model in facial expression recognition[J]. *IEEE Access*, 2019, 7: 184599-184610.

[6] QIAO Guifang, HOU Shouming, LIU Yanyan. Facial expression recognition algorithm based on combination of improved convolutional neural network and support vector machine[J]. *Journal of Computer Applications*,2022,42(04):1253-1259.

[7] FAN Rui, CHEN Xiangyuan, WANG Guannan & CUI Yanhui. Unbalanced dataset anomaly detection and classification algorithms. Proceedings of the CSU-EPSA. doi:10.19635/j.cnki.csu-epsa.001122

[8] M. Saleem Abdullah, S.; Abdulazeez, A. M. . Facial Expression Recognition Based on Deep Learning Convolution Neural Network: A Review. jscdm 2021, 2, 53-65.

[9] JIN X Z, LIN F, WANG Y. Facial expression recognition algorithm based on CNN[J]. Journal of Qilu University of Technology, 2021, 35(3) : 64 − 69

[10] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

[11] J. Nagi et al., "Max-pooling convolutional neural networks for vision-based hand gesture recognition," 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, Malaysia, 2011, pp. 342-347, doi: 10.1109/ICSIPA.2011.6144164.

[12] Ma, W., & Lu, J. (2017). An equivalence of fully connected layer and convolutional layer. arXiv preprint arXiv:1712.01252.