Impact of model extraction information on classification

Bowen Tang

Research School of Computer Science, Australian National University, Canberra Australia
u6726602@anu.edu.au

Abstract. The subject of expression recognition has been studied for many years. There are many uncertainties in the real world, such as pose, exposure, brightness, etc., which can affect the accuracy of expression recognition. Each model has its own unique characteristics. Some studies have shown that many models can achieve good performance in expression recognition after removing the complex and irrelevant factors. In order to find a model that can better adapt to complex uncertainties, we investigate the performance of Convolution structure and Cascade structure on expression recognition respectively. By testing the classification on the same SFEW dataset, the Convolution structure can average get 23.93% accuracy far better than Cascade structure's 19.95%. And Convolution structure's standard deviation is 3.11 has better robustness than Cascade's 6.23.

Keywords: Faces-Emotion Classification · CNN · Casper · SFEW dataset.

1 Introduction

The complex diversity of the environment makes facial expression recognition in the wild very challenging. The currently existing expression classifiers can achieve the desired situation in specific situations. However, the classification results are not satisfactory when extreme SFEW datasets[1] are encountered. Nowadays, expression recognition has many applications in the field of human-computer interaction, such as affect analysis, and mental health assessment. The research on facial expressions in images has recently attracted a lot of interests[5]. In general, emotions include angry, disgusted, fear, happy, neutral, sad, and surprise. Facial expressions can best represent a person's emotions.

To achieve high accuracy face emotion recognition is a computationally intensive and very challenging task. With the recent development of deep learning and parallel algorithms, classification with convolutional neural networks has made great achievements[2]. Deep learning differs from traditional machine learning algorithms in that they are able to perform feature extraction and classification simultaneously. Another advantage of deep learning is that the parameters are updated by back propagation by calculating the error from the ground truth. Therefore, the algorithm can extract unexpected features from the original data. Convolutional neural networks are particularly suitable for 2D image training sets. Convolutional neural networks are essentially a special kind of multilayer perceptron(MLP)[6] .

The Cascade framework is a dynamic and adaptive structure[4]. Different from other frameworks, it determines in real time whether additional neurons are needed for feature extraction and accelerated convergence based on experimental errors. Therefore, the final generated network has no redundant neurons. This helps it to save a lot of computational resources. We all know that a single image contains a huge amount of data information. When we need to train hundreds of images, the amount of input data will be extremely large. If the network framework is pre-defined at the beginning, there is a certain probability that the network framework will not match this huge amount of data. In other words, we will not get good training results. Therefore, it is feasible to use Cascade framework for feature extraction and classification of large number of images.

In this paper, we will compare the performance of the Convolution structure [3] and the Cascade structure[4] for facial expression recognition. First will introduce the SFEW dataset used for our experiments, and Section 2 introduces the characteristics and parameter settings of the Convolution structure and Cascade structure respectively. Section 3 will discuss the results of the analysis experiments. Finally, it is summarised that the Convolution structure has better performance in facial expression recognition.

1.1 SFEW Dataset Inspection and preprocessing

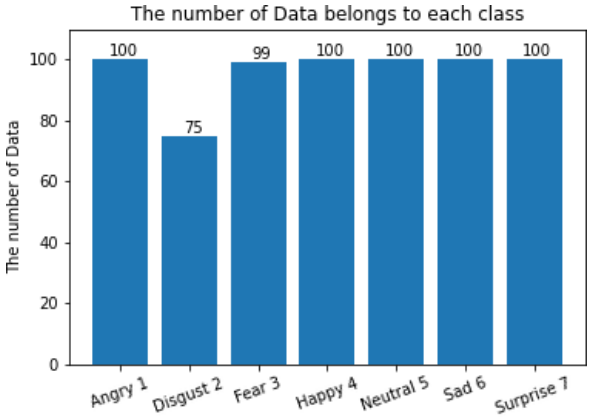
Static Facial Expression in the Wild (SFEW) dataset[1] contains the 10 PCA of 674 images and each of them has been labeled for one of their seven different emotions(Angry, Disgust, Fear, Happy, Neutral, Sad, and Surprise) the

Column 1 in Fig.1(a). Those images are some extremely high or low-resolution faces, which are very close to the real world. The collection of the SFEW dataset has the First 5 PCA for Local Phase Quantisation(LPQ) (Column 2 to 6 in Fig.1(a)) and the First 5 PCA for Pyramid of Histogram of Oriented Gradients(PHOG) (Column 7 to 11 in Fig.1(a)) separately. Using these 10 features to classify the emotion group. The classification accuracy of only using LPQ is 43.71% and only using PHOG is 46.28% mentioned on the paper[1]. The paper also classified that using the current state-of-the-art methods does not perform very well on more real-world conditions. Therefore, train this dataset can ensure the robustness of faces-emotion classification but harder to do classification.

Compared with the features data scale, the difference between the max value and min value is very large see in Fig.1(a), but the mean of each feature class is close to 0 and all of them are on the same scale. Therefore, the normalization of this dataset is not very necessary. Preprocessing only needs to remove the NaN, missing, value and convert all data to float type for computation.

	1	2	3	4	5	6	7	8	9	10	11
count	674.000000	674.000000	674.000000	674.000000	674.000000	674.000000	6.740000e+02	6.740000e+02	6.740000e+02	6.740000e+02	6.740000e+02
mean	4.075668	-0.001475	0.000009	0.000024	-0.000001	-0.000010	6.676558e-10	-1.839763e-09	1.038576e-09	4.243323e-09	-3.709199e-09
std	2.001166	0.010869	0.015953	0.010653	0.007924	0.005979	7.062467e-03	6.493357e-03	5.111828e-03	4.738708e-03	4.681640e-03
min	1.000000	-0.009394	-0.043786	-0.035967	-0.021130	-0.017691	-1.992600e-02	-1.427600e-02	-1.502100e-02	-1.585500e-02	-1.185700e-02
25%	2.000000	-0.003996	-0.011210	-0.006704	-0.005569	-0.004239	-5.323100e-03	-4.799950e-03	-3.573025e-03	-2.827050e-03	-3.113250e-03
50%	4.000000	-0.002518	-0.002169	-0.000359	-0.000430	-0.000187	3.868050e-04	-1.641550e-04	2.360000e-04	1.380850e-04	-1.621050e-04
75%	6.000000	-0.001143	0.009221	0.004986	0.005043	0.003978	5.712225e-03	4.724075e-03	3.645725e-03	2.734750e-03	2.717700e-03
max	7.000000	0.165250	0.054775	0.062508	0.035697	0.017389	1.562200e-02	1.779100e-02	1.395200e-02	2.039900e-02	1.613000e-02

(a) SFEW dataset description



(b) SFEW Distribution

Fig. 1. SFEW Dataset

2 Methodology

2.1 Convolution Structure

CNN is a representative model of Convolution. Convolution of several features of a sample to obtain new features is the characteristic of the Convolution structure. Generally, there will be multiple features to decide the sample belongs to which class. Extracting some connections inside those features could help to gain deeper relationships between features and classes.

Normally CNN will have three parts, Convolutional Layer, Pooling Layer, and Fully-connected Layer see Fig.2. Convolutional Layer can shrink the image scale. An image has a lot of similar areas. What the classification needs

to do is to match a similar pattern in another region. We can convolve those repeated areas into a smaller region. Therefore, the smaller region contains more compact information. From here we can learn some useful experience, applying the convolution operation to the SFEW data set. Squeeze the original data to a higher dimension, which can establish some hidden relationships among the original data. Pooling Layer means downsampling of an image. Take the MaxPool1d as an example, it only preserves the largest value of a specific region, and the value could be positively related to the probability of the right class. Thus, Pooling only reduces the computation parameters and pooled data's structure is still stable. Finally, Fully-connected Layer takes the result of all previous layers to generate the output. Multi-Layer Perceptron(MLP) can be assumed a Fully-connected Neural Network. Take the input and try to find the best parameters to do the classifications.

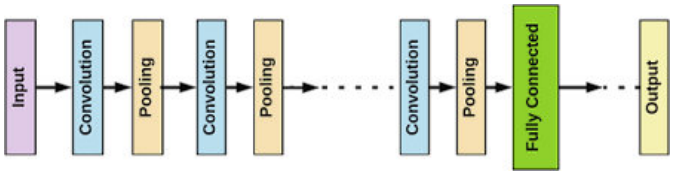


Fig. 2. CNN structure

In a summary, the convolutional structure can generate some high-dimensional data from the original data to train the model. When we combine CNN and MLP together, it can extract the relationship between classes and features from both high dimension and original data. We will compare them with the following structures in order to test the performance of classification.

2.2 Cascade Structure

Casper is a representative model of Cascade. Only inserts a group of hidden neurons into the structure when necessary is the one attribute of the Cascade structure see Fig.3. It can help Casper to construct a proper topology without any redundant neurons and have a better performance. In order to gain as many details as possible from the entail topology, previously added hidden neurons still keep updating and no one is frozen. The input size of the following new neurons will be extended by the output of the previous neurons.

Setting different learning rates for the 3 specific parts is the second attribute of the Casper see Fig.3. The weights have a connection with the newest neuron use L1. The weights of the newest neuron connected to the output layer use L2. The rest of the weights use L3. And the relationship between them is $L1 \gg L2 > L3$. The largest L1 ensures the new hidden neuron learns more information from the previous network. Using a larger learning rate before the output layer can reduce the error of classification. For example, 0.2, 0.005, and 0.001 are suitable learning rate settings for three sections.

2.3 Optimisation

Due to the total dataset has 674 samples for 7 classes and each class has nearly 100 samples see Fig.1(b), all models will set 200 as the batch size which can make sure adequately cover the data of seven classes in each step. All the models will train for 1500 epochs to try to get coverage.

CNN model setting In order to determine which class the data belongs to, more relationships between LPQ and PHOG with the class are needed to be found. Convolution between 10 features of data or between features of data of the same sentiment group can obtain a deeper relationship that how the features determine the same class. Thus, we plan to squeeze each data from the original 10 dimensions into 5 dimensions and do the batch norm to the squeezed data. The main reason is that each data has two PCA and each of them is 5 dimensions. Therefore, try to squeeze them into 5 dimensions could find new relationships inside them. Before entering the next convolution layer, apply the RELU activation function to the result and choose the maximum value by the Pooling layer. RELU activation function only keeps those values that have strong generalisation. MaxPool1d only remains the largest

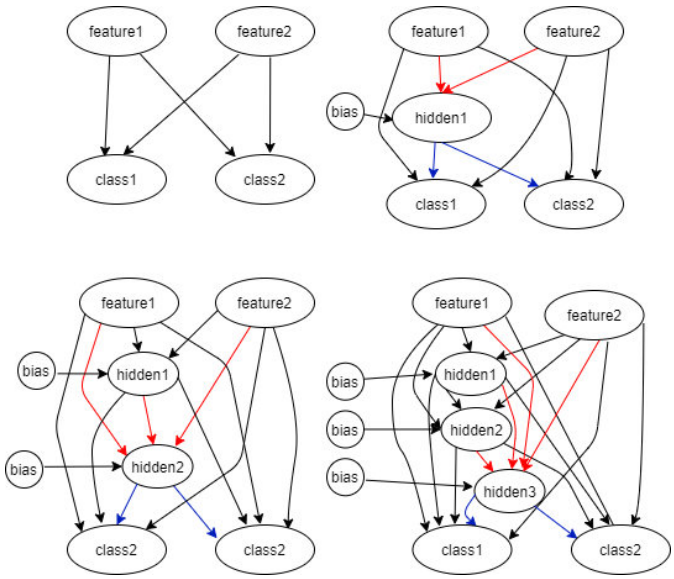


Fig. 3. Cascade Structure. Red lines are Connections with L1, blue lines are Connections with L2, Rest are Connections with L3

value of a specific region and the value could be positively related to the probability of the right class.

Then we will do a similar operation but squeezing the data into 3 dimensions, hope to find new hidden relation. Finally using a fully connected layer to scale the output to seven class dimensions. And apply the F_sofmax to construct the final output. Then use the F.nll_loss to calculate the loss between output and the ground truth. Based on the difference to do the back_propagation and update the parameters by Adam optimizer. The learning rate is 0.005 and the weight_deacy is 0.001.

Casper model setting The local minimal problem is one of the universal problems when using gradient descent to calculate the best hyper-parameter or weights for a model. Exponential learning rate decay [9] can achieve faster convergence and prevent stuck in local minimal.

$$lr = lr * 0.9^{epoch/decay\_step} \tag{1}$$

Three hidden groups can have more than one hidden neuron. The more hidden neurons installed, the more information can be extracted from the data set potentially. With the more hidden neurons installed, the connection inside the neuron network is much larger. Therefore, based on the inserting strategy[7], we are going to check whether the new extra hidden is needed or not when the epoch is satisfied the Eq.(2). Set P a little larger to get a larger checking period in order to make training well-balanced and then install a new neuron. During the gap, the learning rate will be decayed by Eq.(1). If we check the loss too frequently, the operation will make all previous decaying achievements useless. For example, we set P as 10 when hidden neurons less than 5, set P as 15 when the number is less than 10, and 20 when the number is less than the maximum size of 20.

$$epoch = epoch + 15 + N * P \tag{2}$$

Each neuron will take its input value through the Sigmoid Activation function and through the F.log_sofmax to construct the output. Then use the F.nll_loss to calculate the loss between output and the ground truth. Based on the difference to do the back-propagation and update the parameters by Rprop optimizer.

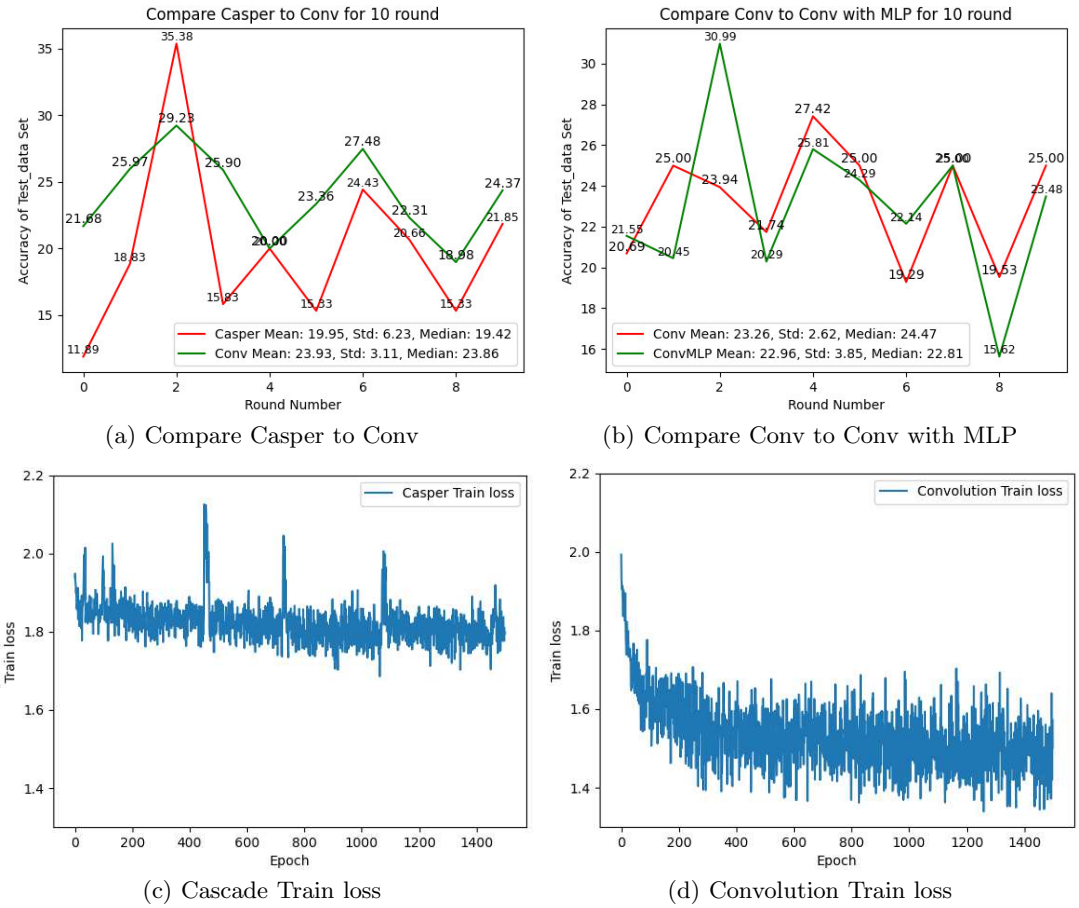
3 Results And Discussion

In this section, we conduct a series of experiments to evaluate the performance of the proposed theory at the beginning. To be more specific, we use the Train_data set to train the neuron network and then apply the network to the Test_data set we have split at the beginning. And the test accuracy will be calculated by Eq.(3). Also, we

will use the LPQ and PHOG two PCA together to train three models. The results will also be compared with the previous researches.

$$Test\ Set\% = \frac{the\ number\ of\ correct\ classification\ result}{the\ total\ number\ of\ Test\_data\ size} \tag{3}$$

Test setting First of all, randomly split 80% data to train the model and the left 20% to test the model at the beginning of each round, the models will not access the test set during the training. Then, using the same Train_data set to train the two models(Cascade with Convolution and Convolution with Convolution + MLP), which will neutralise the influence of random seed, so does the Test_data set. We have compared the representative model of two structures and two different Convolution Structure models for 10 rounds see Fig.4(a) and Fig.4(b) separately.



(a) Compare Casper to Conv

(b) Compare Conv to Conv with MLP

(c) Cascade Train loss

(d) Convolution Train loss

Fig. 4. (a) Compare the Cascade Structure with Convolution Structure (b) Compare two different Convolution Structure (c) Cascade training loss (d) Convolution training loss

Table 1. Compare Three models with SOTA on the same SFEW datasets

	Casper's Test Set%	Conv's Test Set%	ConvMLP's Test Set%	SOTA's Test Set%
Average	19.95	**23.93**	22.96	**56.4**
Std. Dev.	6.23	**3.11**	3.85	
Median	19.42	23.86	22.81	

Three models Comparison Although this is a multi-classification problem, the test set is randomly sampled from the seven classes of data, so the testing can satisfy the class distribution. From Fig.4(a) we can see that Convolution Structure using convolved data has better performance and better robustness when dealing with different test groups. From Table.1, we can see the standard deviation of the Convolution Structure is **3.11** less than the Cascade Structure's **6.23**. Thus Convolution structure is more stable. Although Cascade Structure can get higher peak accuracy **35.38%** it has a more erratic performance situation. When we combined the convolved data with the original data, the green line in Fig.4(b) performances a little bit higher instability, its standard deviation is **3.85**, but still much better than Casper. From Fig.4(c) we can see that there are multiple peaks. Based on the characteristics of the Cascade structure, new neurons are added in the place of those peaks. From Fig.4(d) we can see Convolution structure can achieve convergence relatively quickly.

Previous Researches Comparison The Current SOTA RAN's accuracy on the same SFEW dataset we tested is around **56.4%**[8]. It is structured as a combination of VGG16 and ResNet18. Both structures have a much more complex network complexity than our models. In principle, more complex networks are able to extract more information from the same data, but there is no direct connection. RAN also employs a range of optimisation methods. Such as residual optimisation method and so on. This might be one of the factors for its high accuracy.

```
Casper: feature_size = 10, target_size = 7
train epoch_num = 1500 with batch_size = 200
Casper TRAIN:
Accuracy: 26.70 %
Confusion matrix:
tensor([[15., 11., 13., 15., 12.,  8.,  1.],
        [ 8., 13.,  8., 16.,  8., 11.,  3.],
        [11.,  5., 27., 13., 12.,  4.,  5.],
        [ 6.,  6.,  6., 37.,  8.,  9.,  5.],
        [10.,  4., 11., 14., 26.,  8.,  6.],
        [ 9.,  7., 10., 20., 17., 12.,  6.],
        [ 8.,  7., 13., 14.,  9., 10., 11.]])
Casper TEST:
Accuracy: 23.97 %
Confusion matrix:
tensor([[ 4.,  5.,  6.,  2.,  2.,  5.,  1.],
        [ 1.,  0.,  2.,  0.,  1.,  3.,  1.],
        [ 3.,  3.,  3.,  2.,  3.,  5.,  3.],
        [ 1.,  1.,  4., 12.,  3.,  2.,  0.],
        [ 2.,  3.,  3.,  3.,  7.,  3.,  0.],
        [ 0.,  4.,  2.,  6.,  1.,  5.,  1.],
        [ 2.,  3.,  9.,  1.,  3.,  6.,  4.]])
```

```
Conv: feature_size = 10, target_size = 7
train epoch_num = 1500 with batch_size = 200
Conv TRAIN:
Accuracy: 42.23 %
Confusion matrix:
tensor([[52.,  5.,  3.,  4.,  4.,  6.,  1.],
        [33., 13.,  1., 10.,  3.,  5.,  2.],
        [25.,  1., 36.,  7.,  2.,  4.,  2.],
        [24.,  1.,  3., 44.,  0.,  2.,  3.],
        [28.,  1.,  7., 14., 25.,  4.,  0.],
        [19.,  3.,  3.,  7.,  9., 38.,  2.],
        [34.,  5.,  7.,  6.,  2.,  3., 15.]])
Conv TEST:
Accuracy: 25.34 %
Confusion matrix:
tensor([[14.,  2.,  2.,  0.,  3.,  3.,  1.],
        [ 4.,  0.,  1.,  1.,  0.,  2.,  0.],
        [13.,  4.,  2.,  1.,  0.,  1.,  1.],
        [ 7.,  1.,  2.,  8.,  3.,  2.,  0.],
        [ 7.,  2.,  2.,  4.,  5.,  1.,  0.],
        [ 7.,  2.,  0.,  2.,  2.,  5.,  1.],
        [ 9.,  2.,  5.,  5.,  2.,  2.,  3.]])
```

Fig. 5. Confusion matrixes for Cascade and Convolution structure

Also, by analysing the confusion matrix see Fig.5, Train accuracy of Conv is **42.23%** and its test accuracy is **25.34%**, Train accuracy of Casper is **26.70%** and its test accuracy is **23.97%**. We can conclude that the convolutional structure is more prone to overfitting. The convolutional structure misidentifies more samples into the first class, which leads to an overfitting situation. However, the Casper structure does not. From what we know, convolution is more sufficient in extracting the relationship between features, and thus more samples of the same class can be identified to the correct class, but it also increases the probability of incorrectly identifying other samples to this class. CNN has not very good generalisation capability. Casper, on the other hand, has a similar accuracy in the training and test sets, which means the non-redundant topology is useful.

Overall, the major reason for the low test accuracy is that the SFEW data set is close to real-world conditions. Even we used the PCA of LPQ and PHOG together, the real-world condition is still very complicated. Due to the adaptation attributes of the Cascade structure, the Non-redundant topology sometimes can have very high

classification accuracy. Some experiments show that the Convolution structure could distill more helpful information to performance stability. All of our structures have not enough layers and parameters compare to the SOTA model, which could be one of the drawbacks.

4 Conclusion and Future Work

This paper compares the classification performance of Cascade structure and Convolution structure on the SFEW dataset. Through three sets of model tests, we find that the accuracy and robustness of the Convolution structure are better than those of the Cascade structure on classifying the SFEW dataset. Even though Cascade can occasionally achieve high classification accuracy through its adaptive topology feature, it rarely happens with this high accuracy. For the convolutional structure, since the classification is done by analysing the relationship between features, it is more prone to overfitting.

There is still a lot of potential drawbacks that can be optimised on our Convolution structure. Try to scale up the network's size to obtain more subtle and more obscure relations. Theoretically and technically, the larger size of a network will have more parameters, which means easier to form a more precise model to do the classification. The current SOTA used VGG16 and ResNet18. And it performs far better than ours. Nevertheless, the larger scale of a structure also means the much more parameters should be converged. Thus, we will encounter a new efficiency problem. Extract the face features from the original image could be our next step. Replace 10 PCA with the entire facial features.

References

1. Dhall, A., Goecke, R., Lucey, S., Gedeon, T.: Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In: IEEE International Conference on Computer Vision Workshops (2011)
2. Goodfellow, I.J., Erhan, D., Carrier, P.L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.H., et al.: Challenges in representation learning: A report on three machine learning contests. In: International conference on neural information processing. pp. 117–124. Springer (2013)
3. Jogin, M., Mohana, Madhulika, M.S., Divya, G.D., Meghana, R.K., Apoorva, S.: Feature extraction using convolution neural networks (cnn) and deep learning. In: 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT) (2018)
4. Khoo, S., Gedeon, T.: Generalisation performance vs. architecture variations in constructive cascade networks. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) Advances in Neuro-Information Processing. pp. 236–243. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
5. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25**, 1097–1105 (2012)
6. Shin, M., Kim, M., Kwon, D.S.: Baseline cnn structure analysis for facial expression recognition. In: 2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN). pp. 724–729 (2016). https://doi.org/10.1109/ROMAN.2016.7745199
7. Treadgold, N.K., Gedeon, T.D.: A cascade network algorithm employing progressive rprop. In: Mira, J., Moreno-Díaz, R., Cabestany, J. (eds.) Biological and Artificial Computation: From Neuroscience to Technology. pp. 733–742. Springer Berlin Heidelberg, Berlin, Heidelberg (1997)
8. Wang, K., Peng, X., Yang, J., Meng, D., Qiao, Y.: Region attention networks for pose and occlusion robust facial expression recognition. IEEE Transactions on Image Processing
9. You, K., Long, M., Wang, J., Jordan, M.I.: How does learning rate decay help modern neural networks? (2019)