Multiscale Very Deep Convolutional Network for Facial Expression in the Wild

Xintao Xiang

Research School of Computer Science
Australian National University
u7026949@anu.edu.au

Abstract. In the recent years, with the development of artificial intelligence, more attention has been lay on tasks about human-computer interaction and emotion recognition is one of the popular tasks. However, facial expression recognition (FER) is a very challenging task due to the varying poses under different conditions. The difficulty comes from the fact that some facial expressions are usually close but show completely different meanings. In this paper, I propose a deep convolutional neural network to analyze facial expression and use tricks such as label smoothing and output mix-up to improve the final result. This network combines outputs from different scales of feature maps and is able to achieve 54.32% validation accuracy on the Static Facial Expression in the Wild (SFEW) [1].

Keywords: facial expression recognition, SFEW, Inception Resnet, convolutional neural network, output mix-up, label smoothing

1 Introduction

Face emotion takes an important role in communicating and human-computer interaction. Over recent years, facial expression recognition (FER) has received significant interest from computer scientists. It has widely potential applications such as security, medicine and entertainment and it has been applied to some real-world engineering tasks. However, the analysis of the facial expressions is complex. For example, the eyes may be masked by hands, sunglasses or hairs and mouth may be masked by food or hands. Moreover, the performance is sometimes restricted by the pose of the face like the angle of the face to the camera. These factors make FER a challenging task.

In the early years, many datasets were created for the purpose of facial expression analysis such as CK+ [2], MMI [3], and Oulu-CASIA [4] while most of them are taken in a controlled environment. The control environment means that the expressions mostly are made by humans intentionally. The expressions are not natural. Models work well on these datasets may not have a good generality in real-world environment. To overcome this disadvantage, datasets that include images in uncontrolled environments appear. In this paper, I use Static Facial Expression in the Wild (SFEW) [1] as the evaluation dataset. This dataset captures images from movies. As the (good) actors always attempts mimicking real world human behavior in movies (see Fig. 1), this dataset is close to the real-world environment.

Over the years, a number of deep learning neural networks appear to solve complex computer vision tasks. A number of state-of-the-art methods have got amazing results on specific tasks. Convolutional neural network (CNN) is a very effective method to recognize face expressions [5]. Unlike traditional methods which require users to hand-design image preprocessing methods (POG, Gaussian filtering, etc.) to detect features, this kind of neural network can automatically process and summarize the features. A deep convolutional network can even discover some kind of very high-level representations of features. As deep CNN has been proven to be excellent in image tasks, I expect it to perform well in FER tasks.

In this paper, I propose a deep convolutional neural network that achieves state-of-the-art good performance on SFEW dataset. The proposed network uses transfer learning by using the network described in [6] which is Inception Resnet v1 [7] and pretrained on VGGFace2 [8]. The pretrained net is used to extract the very high-level feature representations of the images. I only use the Inception net before the fully connected layers and add three more convolutional layers after the Inception net. The three layers as well as the Inception net are fully connected together where we take inputs in different scales together.

To get a better result, the model uses the pretrained multitask cascaded convolutional network (MTCNN) [9] to capture the face in the images so that our model can focus more on captured images rather than the noisy backgrounds. Considering the difficulty of analyzing face expressions, several tricks are added to improve the result. I assume that the faces with surprise, angry, happy and other expression are all based on the faces with expression neutral and can be expressed as part of neutral faces plus part of their own. This assumption makes the network better represent the images. Label smoothing [10] is applied to avoid over-fitting and improve generality as face expressions are usually similar to each other. The net is then tested on SFEW [1] dataset and gets 54.32% best accuracy. I compare the performance under different experiment conditions and some other state of the art results on the dataset and it will be analyzed in section 3.

Fig. 1. Example images of different facial expressions in SFEW. Apparently, some of these images are even taken in some complex or even extreme situations (dark for example). This kind of complexity makes the dataset close to real world applications. In addition, as actors try to mimick real-world behaviours, these facial expressions are close to real.

2 Method

2.1 Neural Network

The classification network is based on the Inception Resnet v1 whose details can be found in [7]. I did not make any change to the network structure. Inception was first proposed in [11] and is a successful deep learning neural network architecture and has been successful applied to many computer vision classification tasks. In traditional convolutional networks before that time, network usually uses fixed reception field in one hidden layer. However, in Inception, one block can take several different scales of the inputs. This is because in Inception blocks, taking Inception-A block in Inception-v4 [7] as an example, we have 1x1 convolution filter, 1x1 convolution filter followed by one 3x3 convolution filter, 1x1 convolution filter followed by two 3x3 convolution filters and a pooling layer with one 1x1 convolution filter. These four inputs are then concatenated together. As introduced in [12], two 3x3 convolution filters can be regarded as having reception field of 5x5 and three 3x3 convolution filters can be regarded as having reception field of 7x7, we can see that this kind of block takes different scales of input images and thus can represent more complex data structures.

For Inception Resnet, the network takes ideas from Resnet [13] where we directly take last layer's output and add it into current layer's output before activation. This network has been proven to have better performance than single Inception. The Inception Resnet can achieve higher accuracy in fewer epochs [7].

The face detection network is based on MTCNN introduced in [9]. This network uses three nets called Pnet, Rnet and Onet. Given a train image, we resize the image based on a prechosen resize factor (0.7-0.8 usually). Then we get images with original size, original size x factor, original size x factor x factor and so on. We then input the resized images into Pnet and chooses a fixed number of images based on IOU and classification score. The chosen images are then fed into Rnet where we crop the images based on the coordinates from Pnet, adjusts to more accurate box coordinates, again chooses images based on scores and output. The Onet takes outputs from Pnet and somewhat repeats what Pnet does and outputs the final results. In implementation, I choose the detected face with highest score which usually means we got most parts of the face.

As transfer learning is commonly used in deep learning research, I also use transfer learning rather than train a network whole from raw data. I use pretrained MTCNN which is able to successfully detect more than 99% of faces from the input data after experimenting on the dataset. I use Inception Resnet v1 [7] pretrained on VGGFace2 [8] which contains faces varying poses and ages. This pretrained model can already extract face features from complex faces so that we can use the output of this network directly as our input features. It is also a good idea to directly apply this Inception network to

the dataset and fine-tune the model. The result shows that simply use and fine-tune the very deep network gives a good result.

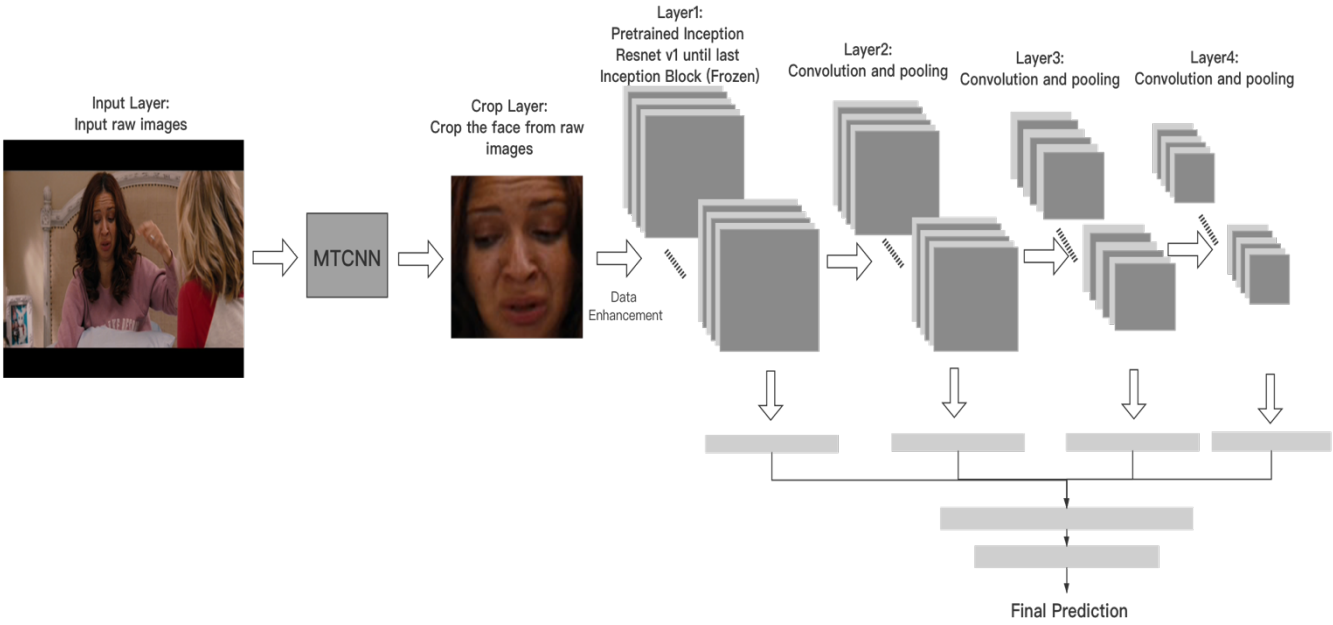

Fig. 2. Neural network architecture of my proposed model. The layer 1 includes layers of Inception Resnet v1 until the fully connected layer and is frozen during the training process. The following three layers contain convolutional filters and pooling to build feature maps of different scales.

Inspired by neural networks that combine outputs from multi-scale reception fields like SSD [14], I designed a neural network that takes inputs from the output of the pretrained Inception net, outputs in four different scales and combines the four scales as shown in Fig. 2. The model frozen the Inception network and uses the outputs from Inception net as the inputs of following three convolutional layers. Each convolutional layer uses different convolutional blocks so that each layer has different reception fields of the images. The outputs of four layers are then concatenated to a same fully connected layer. As the facial expression is usually complex and combination of both large and small actions on face, the model using different scales of outputs can potentially predict the facial expression better.

2.2 Dataset

The dataset is facial expression in the wild (SFEW) dataset [1]. Different from other facial expression datasets that were recorded in a controlled lab environment, this dataset uses facial images extracted from movies. The images are under various real-world environments and the face images are taken under different positions and angles. In addition, as the actors usually try their best in mimicking the differences form real-world behaviors, the facial expressions are closer to the real-world. These features of the dataset make this dataset challenging for recognizing the face expressions.

The dataset contains 675 images where each label angry, fear, happy, neutral, sad and surprise has 100 images while label disgust contains 75 images. The dataset is usually divided into two sets and we use the validation result from training one set and validating on the other set as which also used in [4]. Therefore, we have set 1 which contains 378 images and set 2 which contains 377 images.

2.3 Label Smoothing

In image classification tasks, usually we use one-hot encoding in the label encoding. However, this kind of encoding potentially causes overfitting. Take cross entropy as an example, we optimize

$$l(p,t) = -\sum_{k=1}^{K} t_i \log p_i \tag{1}$$

where t is the label and p is the probability we calculate. This encourages the output scores dramatically distinctive which potentially leads to overfitting [15].

To overcome this problem, the idea of label smoothing was first proposed [10]. It modifies the true distribution of our target values to

$$t_i = \begin{cases} 1-\varepsilon & if\ i = y, \\ \varepsilon/(K-1) & otherwise, \end{cases} \tag{2}$$

where ε is a chosen small value. In this way, we reduce the gap between different sets. This trick has been proven to be useful in image classification tasks and it makes the distribution centers at the theoretical value and has fewer extreme values [15].

In our FER task, we know that face expressions are usually very similar and only have small gaps between each kind of expressions. It is not appropriate to use a hard one-hot encoding to separate each label far away. After experiments about different values of ε, I finally choose 0.3 as my ε.

2.4 Data Augmentation

Data enhancement is widely used in computer vision tasks to improve the generality and avoid overfitting. Considering that we are using a small dataset (fewer than 700 images), we should take measures to enhance the dataset. I use random preprocessing on the input images. In this way, we can actually increase the number of data in our dataset and prevent overfitting on this small dataset.

1. Randomly flip the image horizontally with probability 0.5.
2. Randomly rotate the image by 15 degrees with probability 0.5.
3. Randomly do histogram equalization per channel with probability 0.5.
4. Randomly crop and resize the image to 350-by-350 square image.
5. Randomly cut out 8 holes with maximum size of 6-by-6 in the image with probability 0.5.
6. Normalize the image.

2.5 Output Mix-up

Assumption that every facial expression can be expressed as neutral expression plus its own features is reasonable because as we can imagine, in daily life, we keep neutral expression for most of the time. All our expression can be regarded as an expression that is based on the neutral expression. For example, when we feel happy, a very common action is smiling where we get some angles for the lips and the other parts of the face remain same. This is only a slight change and the model may fail to notice this small change. But if we mix the scores neutral with happy, we can make the model focus more on the change of happy against neutral and potentially get higher accuracy. According to the assumption, every output is mixed-up by

$$y_k = \lambda_5 y_5 + (1 - \lambda_k) y_k \tag{3}$$

where y_k is the kth output and λ_k is the kth mix-up coefficient. In this task, the 5th label is the neutral expression. All the coefficients are set to 0.2 at the initial stage and will be updated during the training. My experiment shows that with label smoothing and output mix-up, the model becomes more stable during training.

2.6 Training Methodology

I divide the dataset by the Stratified 2-fold method. Using this method, the dataset is divided into 2 folds, set1 and set2 which have the same label distribution of labels. I use set 1 to train, test on set 2 and then train on set 2 and test on set 1. I report the best average results on the two test sets.

In training face expression recognition model, it is common to focus on the face rather than introduce noise from the background. In this task, multitask cascaded convolutional network (MTCNN) [6] is used to capture the face. This net is pretrained on face detection and can successfully detect most of faces from raw images. During training, each time we randomly batch-size number of sample images. Then each image is fed into MTCNN to crop the face largest probability. If no face is detected in the MTCNN, then original image is used. The cropped image is then fed into transformer to do data enhancement.

To improve the performance, I use the Focal Loss which was introduced in [16]. Some data in dataset are easy to be classified while the other data are comparatively more difficult to train. This loss gives wrongly classified data a larger weight during the training process so that our model can focus more on the data that are difficult to classify. As we have imbalanced dataset, the focal loss provides a way to set a larger weight to the data whose labels are with smaller number (disgust in our dataset). After experiments, the best result is achieved when all the weights are set the same. A penalizing term for the output mix-up components should be put to constrain the growth or decrease of these terms. However, as I use Adam optimizer in the model with a very small learning rate, the coefficients only change slightly during training. Therefore, the penalizing term is not added.

Changeable learning rates according to some kind of formulas are commonly used in training deep learning neural networks. In training my model, I simply use a naive way where the learning rate decreases by 0.25 of the origin every fixed number of epochs.

Because we do not have a general test set, I report the average validation accuracy on the two sets. For each experiment condition, as the result can vary each time due to the random initialization, I run 3 times of the network and report the average result of the three experiments.

3 Experiment and Result

In this section, I perform my network under different configurations and compare the results. All the hyperparameters for each experiment config, (learning rate, batch size, etc.) are fine-tuned to get best results. I compare my results with other recently developed models such as the model that uses GAN to produce adversary images [17]. I also compare results with one technique [19] that is applied on the SFEW 2.0 dataset which contains more images than my dataset. This comparison is made because I find that though with fewer data, my method achieves result that is close to the result in [19]. The comparisons are displayed in Table 1.

Overall, the networks experimented in this paper have good performance on the dataset and achieve accuracy 20% higher than the model in [17] and are just close to the state-of-the-art model tested on SFEW 2.0 dataset. We can see that the two tricks output mix-up and label smoothing both boost the final results. The output mix-up boosts the single Inception net's final result by 2.1% and label smoothing further improves the accuracy by 0.4%. The best accuracy is performed by my proposed model with output mix-up which achieves 54.32% average accuracy on the validation sets. This high accuracy is because we combine the outputs of different scales. The label smoothing failed in my prosed model. The reason might be that we already mix the output up and the outputs of every label are already close to each other. When we further use label smoothing, the prediction bound of each label is even closer and results in overfitting.

Table 1. Comparison of results under different configs. Texts in Bold represent the model with best performance, texts with underline represent the second-best model.

Config.	Average Acc. %
Baseline by SVM [1]	19.0
DS-GPLVM[18]	24.70
CycleAT [17]	30.75
MTCNN + Inception Resnet v1 (fine-tuned)	51.3
MTCNN + Inception Resnet v1 + Output mix-up	53.65
MTCNN + Inception Resnet v1 + Output mix-up + Label Smoothing	<u>54.19</u>
My Proposed Model + Output mix-up	**54.32**
My Proposed Model + Output mix-up + Label Smoothing	51.48
Covariance Pooling [19]	58.14 (on SFEW 2.0)

Figure 3 shows the confusion matrix of one selected best result (performed by Inception net with all tricks). We can see that the label disgust is most difficult to predict. One of the reasons may be that 'disgust' is very similar to sad and angry. We can imagine that when people feel angry or sad, they may sometimes also feel disgust and take some expression just between disgust and other expressions. We see that disgust is distinguished very well with fear and happy. The label happy is the easiest one to predict.

The loss and accuracy on both training and validation set of the experiment in Fig.3 is shown in Fig.4. From the figure, we can see that our training is stable, the training loss and validation loss can both converge to some values. Though the training loss achieves very close to 100% after 20 epochs, the validation accuracy can still increase a little bit. This is because we have data enhance measure and we are still feeding into some images that the network hardly sees or even never sees. Moreover, as the data is randomly preprocessed, even when we achieve 100% percent accuracy on training set, the network will continue to train as the input images vary each epoch.

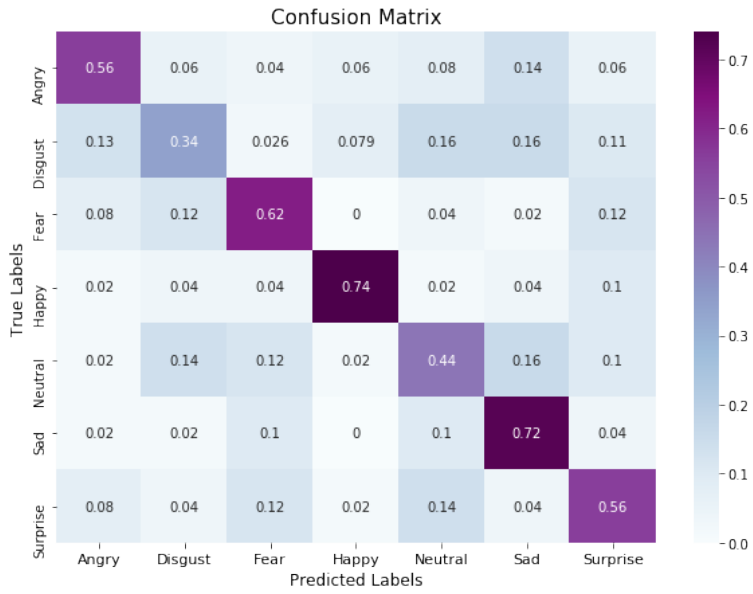

Fig. 3. Confusion matrix for selected best result.

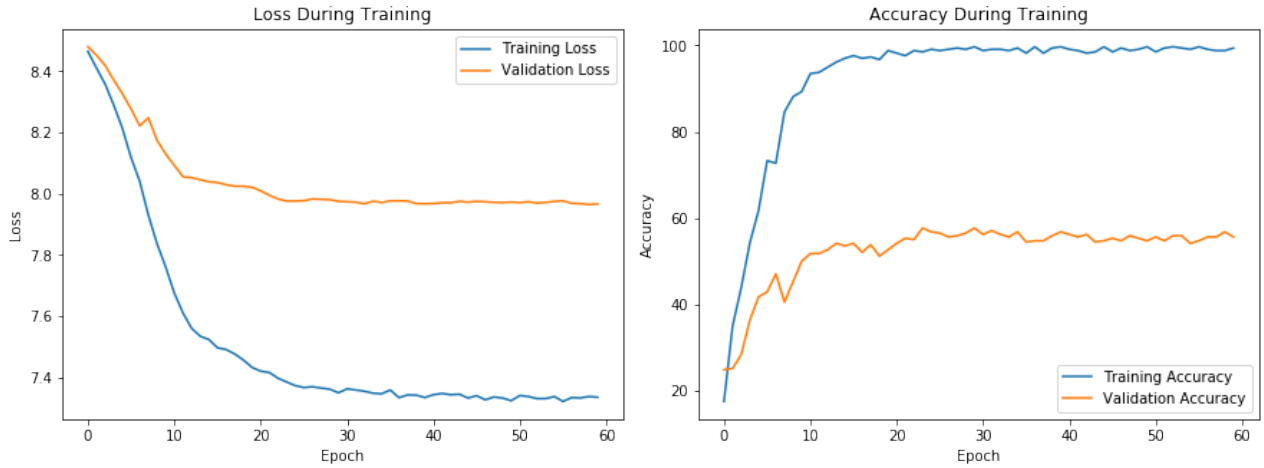

Fig. 4. The figure plots the loss and accuracy on both training set and validation set.

4 Conclusion and Future Work

I apply the very famous Inception Resnet with pretrained weights into facial expression recognition. I also design a network that uses the extracted features from the pretrained Inception network and combines the results from four different scales. The result of my proposed model with output mix-up achieves the best result compared to other models. The fine-tuned model achieves the state-of-art result on SFEW dataset and the result is close to state-of-art result on SFEW 2.0 dataset though with fewer images. The experiments show the power of output mix-up.

My future work will be investigating the effect of output mix-up. I will apply my proposed model to more face recognition datasets like SFEW 2.0 to verify the power of the model on face related datasets. I will try to use fewer pretrained layers from Inception network as the very deep layers have already represented images in a complex space which may affect the performance of different scale outputs. Some advanced loss functions such as Center Loss [20] specially designed for face recognition task are worth for a try. Attention based architecture may further improve the result.

References

1. Dhall, A., Goecke, R., Lucey, S. et al: Static Facial Expression Analysis in Tough Conditions: Data, Evaluation Protocol and Benchmark. In: IEEE International Conference on Computer Vision (ICCV 2011), Curran Associates, Inc., Washington USA, p. 7 (2011)

2. Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., and Matthews, I.: The Extended Cohn-Kanade Dataset (CK+): A Complete Dataset for Action Unit and Emotion-specified Expression. In Proc. CVPRW, Jun. 2010, pp. 94–101 (2010)
3. Pantic, M., Valstar, M., Rademaker, R., and Maat, L.: Web-based Database for Facial Expression Analysis. In Proc. ICME, Jul. 2005, p. 5 (2005)
4. Zhao, G., Huang, X., Taini, M., and Li, S. Z.: Facial Expression Recognition from Near-infrared Videos. Image Vis. Comput., vol. 29, no. 9, pp. 607–619 (2011)
5. Li, H., Lin, Z., Shen, X., Brandt, J. and Hua, G.: A Convolutional Neural Network Cascade for Face Detection. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 5325-5334 (2015)
6. F. Schroff, D. Kalenichenko and J. Philbin: FaceNet: A Unified Embedding for Face Recognition and Clustering. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, pp. 815-823 (2015)
7. Szegedy, C., Loffe, S., Vanhoucke V. and Alemi A.: Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning." ArXiv abs/1602.07261 (2017)
8. Q. Cao, L. Shen, W. Xie, O. M. Parkhi and A. Zisserman: VGGFace2: A Dataset for Recognising Faces across Pose and Age. In 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), Xi'an, pp. 67-74 (2018)
9. K. Zhang, Z. Zhang, Z. Li and Y. Qiao: Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. In IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499-1503 (2016)
10. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna: Rethinking the Inception Architecture for Computer Vision. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 2818-2826 (2016)
11. C. Szegedy et al.: Going deeper with convolutions. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9 (2015)
12. Simonyan, K., and Zisserman, A.: 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, Y. Bengio and Y. LeCun, Eds (2015)
13. K. He, X. Zhang, S. Ren and J. Sun: Deep Residual Learning for Image Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 770-778 (2016)
14. L., Wei & A., Dragomir, Erhan, D., S., Christian, and R., Scott.: SSD: Single Shot MultiBox Detector (2015)
15. Xi, J., He, T., Zhang, Z., Zhang, H., Zhang, Z. and Li, M.: Bag of Tricks for Image Classification with Convolutional Neural Networks (2018)
16. Lin, T., Goyal, P., Girshick, R., He, K.: and Dollar, P. Focal loss for dense object detection. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2999–3007 (2017)
17. Zhang, F., Zhang, T., Mao, Q., Duan, L. and Xu, C.: Facial Expression Recognition in the Wild: A Cycle-Consistent Adversarial Attention Transfer Approach. (2018)
18. S. Eleftheriadis, O. Rudovic and M. Pantic: Discriminative Shared Gaussian Processes for Multiview and View-Invariant Facial Expression Recognition. In IEEE Transactions on Image Processing, vol. 24, no. 1, pp. 189-204 (2015)
19. D. Acharya, Z. Huang, D. P. Paudel and L. Van Gool: Covariance Pooling for Facial Expression Recognition. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, 2018, pp. 480-4807 (2018)
20. Wen, Yandong et al.: A Discriminative Feature Learning Approach for Deep Face Recognition. In ECCV (2016)