Performance of state-of-the-art descriptors on Static Facial Expression Analysis in Tough Conditions: Experiment with BDNN and BPNN

Xinran Tian

Research School of Computer Science, Australian National University
U6920781@anu.edu.au

Abstract. Facial recognition is widely used in real life, and it has received more and more attention from industry experts. People have collected many data sets for facial recognition in the experimental environment, including JAFFE and Multi-PIE, and proved the state-of-the-art descriptors performed well in these data sets. But in real life, facial expression analysis still faces many difficulties. Abhinav and his teammates collected the data set Static Facial Expressions in the Wild (SFEW), which is closer to actual life, and applied support vector machine to classify it. The results prove that the performance of state-of-the-art descriptors such as LPQ and PHOG are clearly not suitable for facial expression analysis in SFEW. In this report, we will use BPNN and BDNN to classify facial expressions in SFEW data set, and re-verify whether LPQ and PHOG descriptors are suitable for SFEW. The experimental results show that although the accuracy of BDNN is slightly better than BPNN, the final results of both are unsatisfactory, which once again confirms Abhinav's conclusion that LPQ and PHOG are not suitable for uncontrolled environments. And PHOG works better compared with LPQ. At the same time, we proved that bidirectional training has a certain degree of optimization effect on the classification accuracy of neural networks.

Keywords: Facial Expression Analysis · BPNN · BDNN.

1. Introduce

Facial expression analysis includes both the measurement of facial movement and the recognition of facial expressions. These facial expressions are produced by changes in human facial muscles, and these changes convey personal influence to the observer.[1] With the development of science and technology, facial expression analysis is widely used in all aspects of life, including human computer interaction (HCI), affective computing, human behavior analysis, ambient environment and smart homes, pain monitoring in patients, stress, anxiety and depression analysis , lie detection and medical conditions such as autism.

Some facial expression analysis methods are based on image [2] and some are based on video [3]. However, sometimes we can't get temporal data, so expression analysis methods become more considerable in life. In our life, there are many typical applications of expression analysis methods based on image, such as smile detection [4] and expression-based album creation [5].

The JAFFE database [6] is one of the earliest static facial expressions dataset. It was collected from 10 Japanese females, each of them was asked to posed for six expressions (angry, disgust, fear, happy, sad and surprise) and the neutral expression. It has been extensively used in many researches[7,8]. However, it was created in a lab-controlled environment, so it doesn't have enough ability to present the real condition in life. Figure1 shows some pictures from JAFFE, which is the evidence of that JAFFE obtained from a lab-controlled environment.

figure 1. Pictures from JAFFE

Multi-PIE [9] database is another popular and widely used database. The CMU Multi-PIE face database contains more than 750,000 images of 337 people recorded in up to four sessions over the span of five months. The main limitation of these databases is also that they have been recorded in lab-controlled environments. Figure2 shows some pictures from Multi-PIE, which is the evidence of that Multi-PIE obtained from a lab-controlled environment.

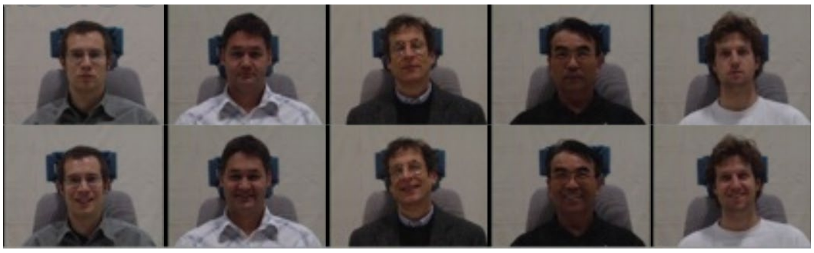

figure 2. Pictures from Multi-PIE

Abhinav and others extracted Static Facial Expressions in the Wild (SFEW) database [1] from the temporal dataset Acted Facial Expressions in the Wild (AFEW). AFEW [10] is a dynamic facial expression database, extracted from the clips of 37 movies, and contains a large age range of subjects from 1-70 years. When choosing movie scenes, people tried to choose realistic scenarios and considered the needs of large age range. Although the films were shot in a controlled environment, they are closer to the real environment. Although the actors also pose in the film, it is clear that good actors will try to imitate the real-world behavior in the film. Therefore, the facial expression data set obtained in the movie is much more realistic than the data set recorded in the laboratory environment. So SFEW provides a much more difficult test set than currently available datasets. Figure3 shows some pictures from SFEW, they are come from several movies and are close to the real life.

figure 3. Pictures from SFEW

Abhinav and his teammates experimented with the pyramid of histogram of oriented gradients (PHOG) descriptor and the local phase quantization (LPQ) descriptors in their experiments. LPQ descriptor is calculated on grids and then concatenated for an image. PHOG is the extension of HOG, which counts occurrences of gradient orientation in localized portions of an image. They are all state-of-the-art descriptors. For classification, they used a support vector machine [11]. The type of kernel was C-SVC, with a radial basis function (RBF) kernel. Their result shows that LPQ and PHOG have high performance accuracy on JAFFE and Multi-PIE but significantly lower accuracy for SFEW, because SFEW is closer to real world conditions than JAFFE and Multi-PIE [1].

In order to reprove the result from Abhinav [1], we will use Back Propagation Neural Network (BPNN) and Bidirectional Neural Networks (BDNN) to classify SFEW with PHOG and LPQ, we will also compare the classification performance among the two NN models.

There are two types of neural networks, one is biological neural network and the other is artificial neural network (ANNs). Artificial Neural Networks are also referred to as neural networks (NNs) or connection models (Connection Model) [12]. It is an algorithm model that imitates the behavioral characteristics of animal neural networks and performs distributed parallel information processing. This kind of network depends on the complexity of the system, and adjusts the interconnected relationship between a large number of internal nodes to achieve the purpose of processing information. In the past ten years, the research work of artificial neural networks has made great progress. It has successfully solved many problems in the fields of pattern recognition, intelligent robots, automatic control, predictive estimation, biology, medicine, and economics. At present, there are nearly 40 kinds of neural network models, including Hopfield Network (HN) [13], Convolutional neural networks (CNN) [14], Generative adversarial networks (GAN) [15], etc.

BPNN is the most basic neural network, and its output is forward-propagated, and its error is back-propagated (BP). BP algorithm is based on the gradient descent method. The input-output relationship of the BP network is essentially a mapping relationship: the function of an n-input and m-output BPNN is a continuous mapping from n-dimensional Euclidean space to a m-dimensional Euclidean space, and the mapping is highly nonlinear.

BDNN [16] is built based on BPNN. In BDNN, input and output are both used to train node weights. In other words, not only the back-propagation from output to the input can modify the weight, but the back-propagation from input to the output direction can also modify the weight. AF Nejad and TD Gedeon [16] confirmed in their experiments that BDNN can be trained as associative memories and cluster centroid finders and are capable of classification or prediction, in real world problems. In this experiment, we will use two fully symmetric BPNNs to implement the BDNN model, so that the not only input but also output can train the neural network.

The experimental results show that, when using LPQ or PHOG as descriptors, neither BPNN nor BDNN can classify SFEW data very well, which once again proves Abhinav s conclusion that LPQ and PHOG are not applicable to uncontrolled environment. In addition, by comparing the performance of the two neural networks in the experiment, we found that bidirectional training is beneficial to improve the accuracy of the neural network.

2. Net Details

BPNN

BPNN has three layers: input layer, hidden layer and output layer. The input layer receives data and the output layer outputs data. The neurons in the previous layer are connected to the neurons in the next layer, collect the information transmitted by the neurons in the previous layer, and pass the value to the next layer after "activation".

The learning process of BP algorithm consists of forward propagation process and back propagation process. In the forward propagation process, the input information passes through the input layer and hidden layer, and is processed layer by layer and passed to the output layer. If the expected output value cannot be obtained at the output layer, the sum of the square of the output and the expected error is taken as the objective function, which is transferred to the back propagation. Calculate the partial derivative of the objective function to the weight of each neuron layer by layer, which form the ladder of the objective function to the weight vector, and used as the basis for modifying the weight. The learning of the network is completed during the weight modification process, when the error reaches the expected value, the network learning ends.

In order to find the inverse function of the activation function to establish the BDNN model, x = y is used as the activation function of the hidden layer and the Sigmoid function is used as the parameter of the output layer in the experiment. Select the optimizer SDG, Learning rate = 0.1, weight_decay = 0.0001.

BDNN

In this experiment, BDNN is implemented with two symmetric BPNNs, which are Forward neural network (ForNN) with attributes as input, 7 labels probability as the output, and Reverse neural network (RevNN) with 7 labels probability as the input and attributes as the output. ForNN is exactly the same as BPNN we used before, and RevNN has the same number of hidden layer nodes as ForNN. For symmetry, RevNNs hidden layer activation function and output layer activation function are the inverse functions of ForNNs output layer activation function and hidden layer activation function, respectively.

In each epoch, ForNN passes the optimized parameters to RevNN, which is then trained by RevNN, and then RevNN passes the optimized parameters back to ForNN. Due to the symmetry of the two neural networks, the parameters need to be transferred after transposition. We use Torch to complete the conversion of parameters (node weights) between the two neural networks. In this way, the neural network is trained once through input and output in an epoch. Figure4 and Figure5 show how to transfer weights between two neural networks by torch.

```
input_to_hidden_weights_fornet = fornet.fc1.weight.clone()
hidden_to_out_weights_fornet = fornet.out.weight.clone()
state_dict = revnet.state_dict()
state_dict['fc1.weight'] = torch.t(hidden_to_out_weights_fornet)
state_dict['out.weight'] = torch.t(input_to_hidden_weights_fornet)
revnet.load_state_dict(state_dict)
```

Figure 4. Pass weights form ForNN to RevNN

```
input_to_hidden_weights_revnet = revnet.fc1.weight.clone()
hidden_to_out_weights_revnet = revnet.out.weight.clone()
state_dict = fornet.state_dict()
state_dict['fc1.weight'] = torch.t(hidden_to_out_weights_revnet)
state_dict['out.weight'] = torch.t(input_to_hidden_weights_revnet)
fornet.load_state_dict(state_dict)
```

Figure 5. Pass weights form RevNN to FroNN

The two neural networks use the same optimizer SDG, Learning rate = 0.1, weight_decay = 0.0001.

3. Experiments

The experimental data set includes two parts, the first part is the first five attributes of the SFEW data set transformed with LPQ, and the other part is the first five attributes of the SFEW data set transformed with PHOG. 20% of the data set is used as the test set, and another 80% of the data set is used as the training set.

Normalization is necessary in Experiments, because it can avoid the gradient explosion and improve the convergence speed and accuracy of the model. Figure6 and Figure7 show the BPNN accuracy on test set before and after doing normalization. (use PHOG as input)

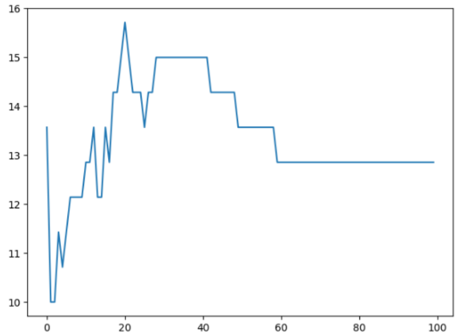

Figure 6. Before Normalization

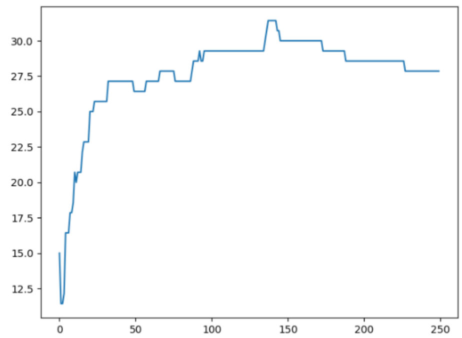

Figure 7. After Normalization

The training set is unbalanced, which affects the accuracy of classification to some extent, so we use oversampling to improve accuracy of model. Figure8 shows the number of each label in training set before and after oversampling. Figure9 and Figure10 show the BPNN accuracy on test set before and after using oversampling. (use PHOG as input)

```
Counter({0: 100, 3: 100, 4: 100, 5: 100, 6: 100, 2: 99, 1: 75})
Counter({0: 100, 1: 100, 2: 100, 3: 100, 4: 100, 5: 100, 6: 100})
```

Figure 8. number of each label in training set before and after oversampling

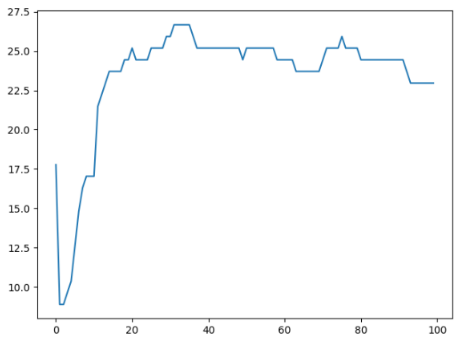

Figure 9. Before Oversampling

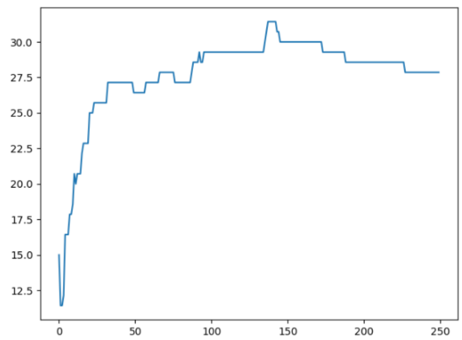

Figure 10. After Oversampling

After normalizing and oversampling the data, the first five attributes of LPQ and PHOG are taken as inputs respectively, and trained through two neural networks. We recorded the classification accuracy of each neural network on the test set.

4. Result and Discussion

When using PHOG as descriptors in SFEW, the classification accuracies on test set are shown in Figure11 and Figure12.

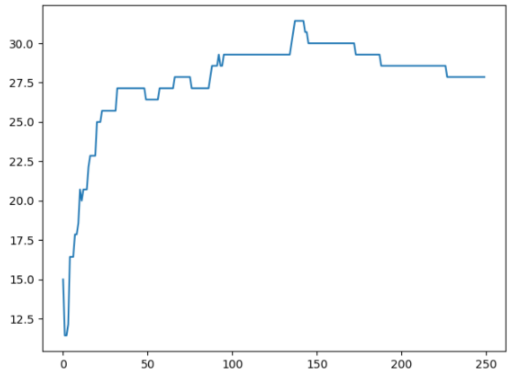

Figure 11. Accuracy on BPNN when using PHOG

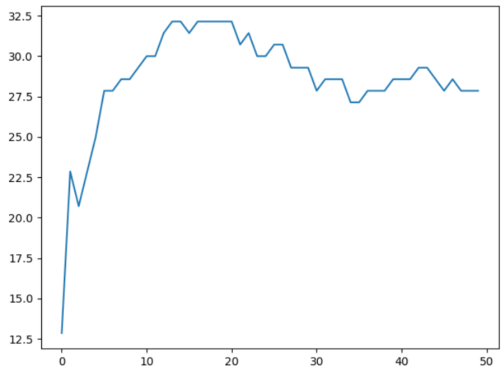

Figure 12. Accuracy on BDNN when using PHOG

When using LPQ as descriptors in SFEW, the classification accuracies on test set are shown in Figure13 and Figure14.

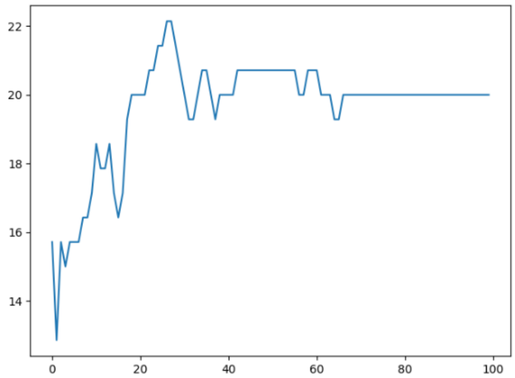

Figure 13. Accuracy on BPNN when using LPQ

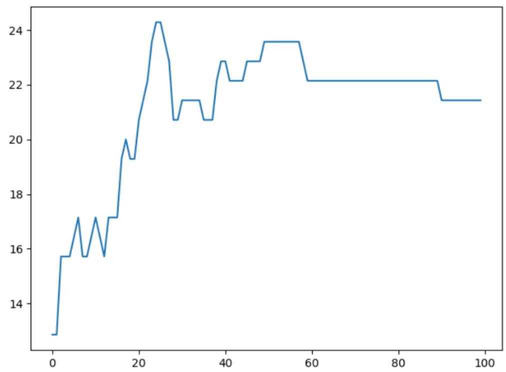

Figure 14. Accuracy on BDNN when using LPQ

When using LPQ or PHOG as a symbol, neither BPNN nor BDNN's accuracy exceed 33%, which shows that the two descriptors are not suitable for such a close-to-life data set, and once again verified the experimental results of Abhinav et al [1].

By comparing the two descriptors, we found that when PHOG is applied, the accuracy can exceed 30%, but when using LPQ, the accuracy is less than 25%, indicating that PHOG is more adaptable than LPQ for the data set SFEW.

By comparing the accuracy of BPNN and BPNN horizontally, we found that whether using PHOG or LPQ, the classification accuracy of BDNN is slightly higher than that of BPNN, which shows that bidirectional training can help improve the accuracy of neural networks.

5. Feature work

This paper only tried two simple neural networks to classify SFEW. We hope to try more neural networks in the future to find neural networks that perform better with the existing descriptors. In addition, we will try to apply bidirectional training on other types of neural networks, not only BPNN, and try to combine other technologies with bidirectional training in order to improve the neural network accuracy. When designing BDNN, we still face many problems, such as the reversibility of the activation function and the classification problem, whether we should use the bias, how to make the network symmetrical under the condition of using bias, and so on. Solving these problems is also part of future work.

As for facial expression analysis field, trying to find new descriptors which are more suitable for facial expression analysis in uncontrolled environment is a important task in the feature.

Reference

[1] Dhall A , Goecke R , Lucey S , et al. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark[C] IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 6-13, 2011. IEEE, 2011.

[2] F. Bougourzi,F. Dornaika,K. Mokrani,A. Taleb-Ahmed,Y. Ruichek. Fusing Transformed Deep and Shallow features (FTDS) for image-based facial expression recognition[J]. Expert Systems With Applications,2020,156.

[3] M. Valstar and M. Pantic. Fully automatic facial action unit detection and temporal analysis. In Proceeding of IEEE International Conference on Computer Vision and Pattern Recognition Workshop, CVPR-W'06, pages 149–149, 2006.

[4] J. Whitehill, G. Littlewort, I. R. Fasel, M. S. Bartlett, and J. R. Movellan. Toward practical smile detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 2106–2111, 2009.

[5] A. Dhall, A. Asthana, and R. Goecke. Facial expression based automatic album creation. In Proceedings of the 17th international conference on Neural information processing: models and applications - Volume Part II, ICONIP'10, pages 485–492, 2010.

[6] M. J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba. Coding facial expressions with gabor wavelets. In Proceedings of the IEEE International Conference on Automatic Face Gesture Recognition and Workshops, FG'98, 1998.

[7] Tanjea Ane,Md. Fazlul Karim Patwary. Performance Analysis of Similarity Coefficient Feature Vector on Facial Expression Recognition[J]. Procedia Engineering,2016,144.

[8] Hyun-Soon Lee, Bo-Yeong Kang. Continuous emotion estimation of facial expressions on JAFFE and CK+ datasets for human–robot interaction. Intelligent Service Robotics, 2020, Vol.13 (1), pp.15-27.

[9] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. In Proceedings of the Eighth IEEE International Conference on Automatic Face and Gesture Recognition, FG'2008, pages 1–8, 2008

[10] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon. Acted Facial Expressions in the Wild Database. In Technical Report, 2011.

[11] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines, 2001.

http://www.csie.ntu.edu.tw/ cjlin/libsvm.
[12] Artificial neural networks, Baidu Encyclopedia, "Science Popularization China" Science Encyclopedia Entry Compilation and Application Project, 2005.
[13] J. J. Hopfield,D. W. Tank. Biological Cybernetics. “Neural” computation of decisions in optimization problems[J] . 1985
[14] Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks, NIPS, 2012.
[15] Lan Goodfellow, Generative Adversarial Networks, ICLR,2014
[16] A.F. Nejad, T.D. Gedeon. BiDirectional Neural Networks and Class Prototypes.