Exploring factors affecting facial expression classification performance with transferred deep neural network

Chaoyin Chen[1]

[1]The Australian National University, Canberra ACT 0200, Australia

U6535292@anu.edu.au

Abstract. This paper is trying to figure out the best settings of the transferred Convolutional Neural Network (CNN) for emotion classification on SFEW in comparison with BDNN. The proposed settings contain a face detection module based on the multi-task CNN (MTCNN), followed by a classification network that was based on the ResNet18 with the last several blocks transferred. All blocks of the network were initialized with the weight that was pre-trained on the larger dataset named ImageNet. Then the model was fine-tuned on the training set of SFEW. The performance was evaluated using the validation and test accuracy and was compared with the performance of Bidirectional Neural Network (BDNN) which was previously experimented. By analyzing the results with different settings, I found that by transferring the last two layers (including 4 convolutional layers and two fully connected layers) of ResNet18, with MTCNN pre-processing the SFEW. the performance of the transferred networks achieves 41.36% on the validation set of SFEW, surpassing the performance of BDNN of 15.92% with significant gains.

Keywords: Transfer Learning, Convolutional Neural Network, Parameter Tuning, Bidirectional Neural Network

1. Introduction

Over the past years, one key problem in human-computer interaction is how to automatically perceive and recognize human emotions [5]. Researches on it involve a wide variety of related fields, including computer vision, speech analysis, linguistics, cognitive psychology, robotics and learning theory, etc. [6]. A computer will understand humans and interact more naturally if it is equipped with a more powerful emotion recognition algorithm. There are also many real-world applications that benefit from this intelligence, including driving safety enhancement and support devices for blind people.

Facial expression of emotion plays a key role in social interaction. They are thought to serve a critical communicatory function between conspecifics [4]. Recent advances in emotion recognition concentrate on recognizing more spontaneous facial expression. The Static Facial Expression in the Wild (SFEW) [3] dataset was collected to simulate spontaneous scenarios. It contains 7 basic emotion categories: anger, disgust, fear, happy, neutral, sad, surprise. The SFEW forms by extracting static frames from Acted Facial Expression in the Wild (AFEW) which is a set of film clips [3]. The idea of SFEW is that, although actors in movies are still not truly spontaneous, it still provides facial expressions much more naturally and versatilely compared to lab-controlled environments. Researchers showed that the model, which had a good performance on the conventional datasets such as JAFFE [7], obtained significantly lower performance on SFEW [3].

Deep convolutional neural network and residual network are often implemented when addressing image classification problems and has recently yielded excellent performance in the wide variety of image classification tasks [1,8,9,10]. The careful design of local to global feature learning with convolution, residual block, pooling and layered architecture provides powerful tools for facial expression recognition. Transfer learning is also often considered because this highly separable architecture makes it easy for modification to fit into similar types of problems. It shortens the convergence time and saved much effort from training from scratch. Many pre-trained models for image recognition could provide a good starting point for transfer learning such as ResNet50, which is integrated image classification and pre-trained on ImageNet [11]. In this paper, I focus on making the best prediction of SFEW with transferring the simplified version of ResNet50, called ResNet18 by testing different settings of the model parameters. The main procedures can be summarized as follows: 1. Configuring the face detection module to provide more accurate detection of faces. 2. Experimenting with different extents of transferring for better performance. 3. Fine-tuning model to fit in the dataset. 4. Evaluating the model performance by validation and test accuracy. By going through those steps, I obtained the best setting: transferring the last two blocks and pre-processing the dataset with MTCNN, the performance of the model obtains 43.24% for the test set, surpassing the 15.92% for the performance of BDNN with a significant gain of 27.32%.

1.1. Residual Blocks

The core of this model involves a residual connection between layers. A residual block is recognized as a basic unit of the residual network. As shown in fig.1 left, if we define the normal feed-forward network as F(x), then what the residual connection does is skipping one or more layers and bring the previous x to later layers. [1] shows that by adding those

[1] Student of Bachelor of Information Technology in Australian National University

residual connections to the network gains improvement in performance by reducing top-1 error by 3.5% on the ImageNet dataset [11] compared with the plain network.

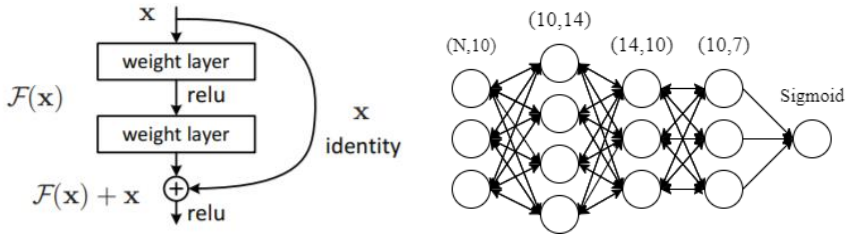

Figure 1. residual block visualization(left). BDNN connection in previous experiment(right)

1.2. Bidirectional Neural Network

The BDNN constructed in the previous experiment consists of three hidden layers as shown in fig.1 right. Apart from the normal feed-forward network, the BDNN contains extra reverse connections between layers, which allows it to remember input patterns as well as output vectors [2]. The forward part of the network shared weight with the reverse part while the bias of two parts are independent to ensure they are training on a different scale. The expected output for the forward part is to do the normal prediction on the SFEW classification and the aim for the reverse part is to enhance features learned from the training dataset [2]. Table.1 demonstrates some details of parameters for BDNN.

Table 1 Detailed parameter settings for BDNN

	Learning Rate	K-Fold	Num_Epoch	Criterion	Optimiser	Input shape	Output Shape
Forward Net	0.03	10	200	CrossEntropy	Adam	(N, M^2)	(7, 1)
Backward Net	0.03	10	200	MeanSquare	Adam	(7, 1)	(N, M)

We could notice that most of the two parts share the same setting except the evaluation criterion. The Backward Net used MSLoss because the output of the net is of shape N×M while CrossEntropyLoss does not support such input size. Besides, MSLoss is enough for enhancing purpose.

2. Method

The transferred model combines different methods to achieve better performance than BDNN, involving processing raw SFEW with MTCNN for face extraction, selecting parameters for the number of transferred layers, random sampling for evenly distributed labels, and K-fold cross validation, etc.

2.1. Resnet18 architectures

Plain Network. The plain baseline is inspired by the philosophy of VGG nets [11] (Figure 2 top). The convolutional layers mostly have 3×3 filters and obeying two simple design rules: for each layer with the same output size, they have the same number of filters; for layers between which the number of feature sizes is halved, then the number of filters is doubled to preserve feature complexity. The network performs downsampling directly by convolutional layers that have a stride of 2. The network ends with a global average pooling layer and a 10-way fully connected layer with SoftMax [1]. For the sake of emotion classification, the output features of the fully connected layer have been modified to 7. One main thing that differs ResNet18 from VGG is that ResNet18 has fewer filters and lower complexity than VGG making it computationally efficient that VGG network.

Residual Network. Based on the plain network, researchers inserted shortcut connections to turn the network into residual version [1]. There are two different types of connections. The identity shortcuts (solid shortcuts in fig.1 bottom) can be directly used when the input and output have the same dimension. When the dimension increases (dashed shortcuts in fig. 1 bottom), the shortcut will perform identity mapping with extra zero paddings. In the latter case, the shortcut performed with a stride of 2 to keep the output size the same as the plain network.

2.2. Transfer learning

Transfer learning is what happens when the model is easier to learn chess when it has already known how to play checkers, or to recognize monkeys when it has learned to recognize gorillas. It has been recognized as an important topic in machine learning research [12,13]. Since the level of learned features increases as the layers go deeper, it would be efficient if we

[2] N refers to the size of the input data and M refers to the number of features for each data point

make use of the pre-trained low-level features and only adjust high-level features on the given dataset, especially when the dataset we are going to train on is relatively small.

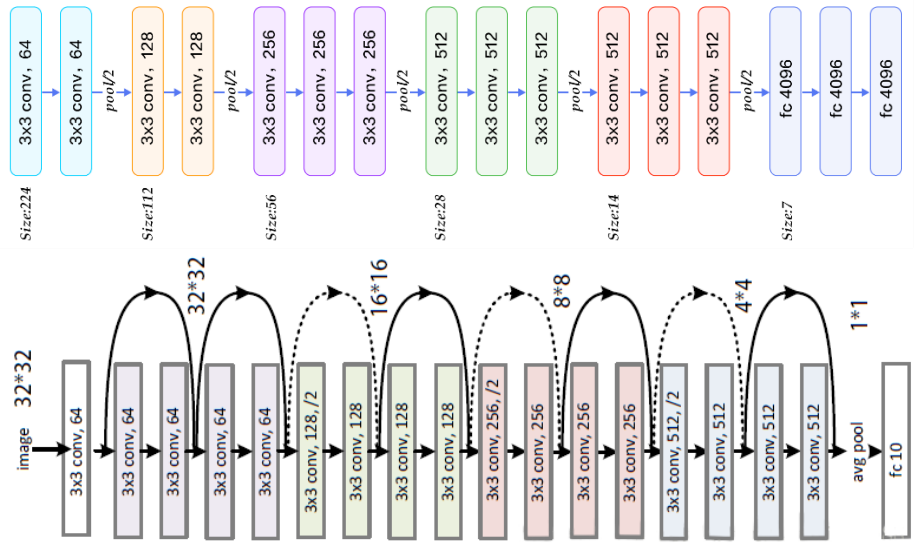

Figure 2 VGG-16 as reference (top). ResNet18 (bottom)

The ResNet18 is pre-trained on the ImageNet, which concentrates on distinguishing between different objects. So, it is important to decide to which extent should we transfer the original network so that it would be more integrated on distinguishing facial expressions. The candidate parameters for blocks to be transferred is 1 to 3. The network would be easy to overfit to the SFEW because of the small number of data points compared with the number of parameters to learn if unfreezing more blocks, or not any high-level features could be extracted if we do not unfreeze enough layers.

2.3. Face detection

To give a more precise prediction on SFEW, MTCNN [14] with pre-trained weights is introduced to help the model concentrate more on distinguishing facial expression instead of wasting memory finding out where the face is. The MTCNN consists of three parts. The first part called the proposal network, which is used to obtain windows and their bounding box regression vectors. Non-maximum suppression (NMS) was implemented at this stage to merge highly overlapped candidates. Then all candidates are fed into another CNN called Refine network, performing calibration with bounding box regression as well as NMS merge. The last part is similar to the previous one but it is aimed to give a more detailed description of faces and output five facial landmarks' positions. By going through this cascaded structure, the MTCNN gradually predict the location of the face in a coarse-to-fine manner. It is an unsupervised procedure so the method used to check whether it is performing well is by randomly pick up samples processed by it. Fig. 3 gives a clear illustration of how detection function works to pre-process the whole dataset.

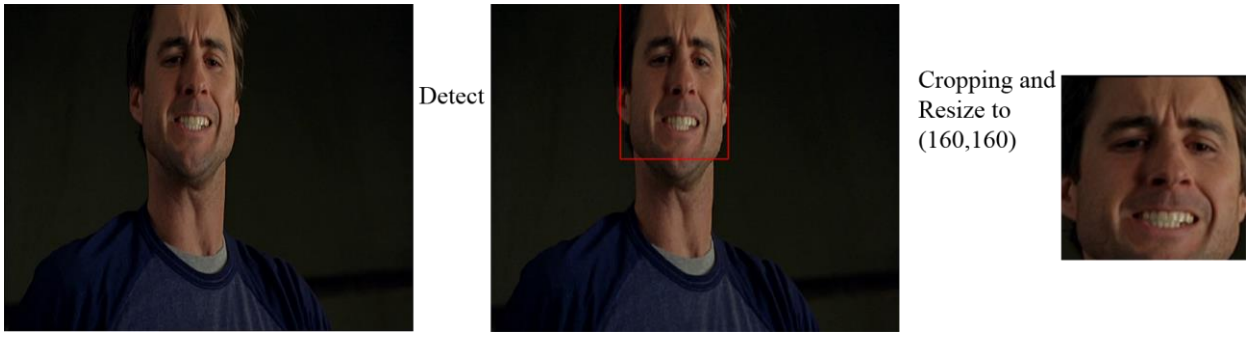

Figure 3 face detection process illustration

Exceptions would happen if the raw images have poor quality so that the MTCNN is not able to distinguish where the face locates. In this case, the raw images would only be resized into (160, 160) and feed into the model. It is generally recognised as the noise in the dataset.

2.4. Other settings

Apart from techniques introduced above, some other methods were also implemented to ensure the robustness of the

training process, including K-fold cross-validation, CrossEntropyLoss criterion and Adam optimizer. K-fold cross-validation divides the dataset into several blocks and split the training process into K iterations. In each iteration, the picked block is treated as validation set and others as training set. It provides an idea of how the model is performed during the training process. It also makes the most of the dataset since we could do the prediction on every data point, which is helpful when the dataset is small. The K is a hyperparameter and is chosen to be 5 by convention. It is implemented in the model by randomly sampling and splitting the dataset according to K for each epoch.

CrossEntropyLoss was selected as a criterion as it is a powerful and conventional tool for classification. Adam [15] is selected as the optimizer. It is an adaptive learning rate optimization algorithm that is integrated for training deep neural networks, and has been proven to be a computationally efficient algorithm for gradient-based optimisation of stochastic objective function [15].

3. Result and discussion

Data were collected based on different transferring layers and before and after applying face extraction.

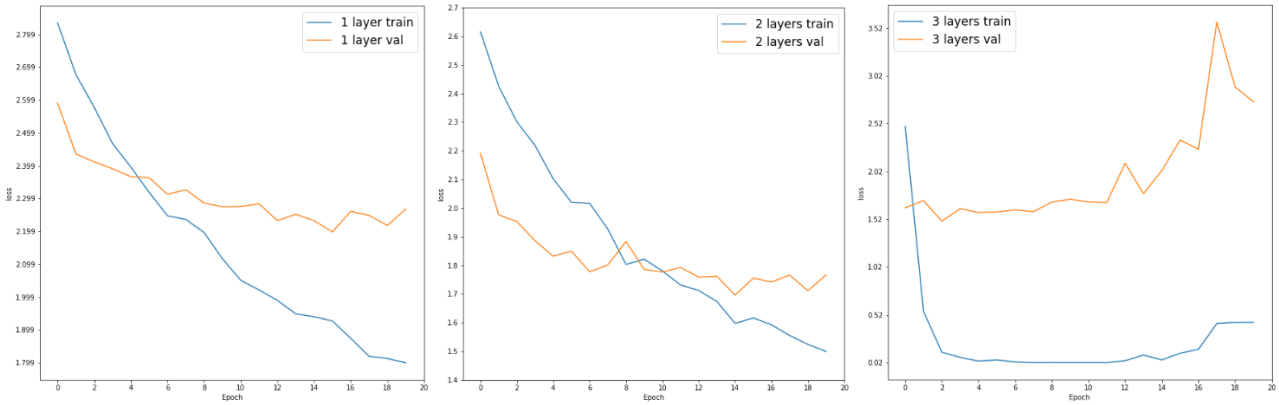

Figure 4 training and validation error when number of transferred layers = 1, 2, 3 (with face extraction applied)

By analyzing the training and validating loss for the first 20 epochs I got the fig. 4. When only one block was transferred (fig.4. left), which is only the last fully connected layer in the ResNet18 structure, we can find that it seemed to have a valid trend for training loss. But one issue was that the overall loss for the validation set is higher compared with that of the 2-layer case. Also, we could notice that the relationship between training loss and validation loss is not so obvious. One potential reason for that is the high-level features learned by the last layers are not stable enough to establish connections with previous layers. When the number of blocks transferred equals three (fig.4. right), we could explicitly find that the overfitting issue occurs. Since three blocks involve 8 convolutional layers as well as the last fully connected layer, the SFEW is too small to help such network learn features without memorizing it. Accordingly, when n = 2 (fig.4. middle), it produces a relatively good prediction, and the relationship between training and validation error could be obtained. Table.2 also gives statistics on training and validating accuracy for those three options. It shows that when n = 2 the model could obtain a relatively good validation accuracy.

No. transferred layers	Training accuracy (%)	Validation accuracy (%)
1	34.42	26.51
2	41.36	43.24
3	87.68	14.42

Table 2 average training and testing accuracy for n layers (n = 1,2,3) in 20 epochs (with face extraction applied)

	Apply extraction block	Not apply extraction block
Average Training loss	1.50663	2.271788
Average Validation loss	1.78958	1.91592

Table 3 Average Loss before and after applying face extraction block

Table.3 shows the average loss for training and validating with and without the face extraction block applied as the pre-processing tool over 20 epochs. We could notice that on average the loss without blocks applied is worse than the loss that the block is applied. This pre-processing helps the model gain a better performance as the loss for validation drops. It also makes the model easier to learn features from the dataset as the average training loss is lower when such a block has been applied.

3.1. Result from BDNN

Below were statistics collected from previous experiment.

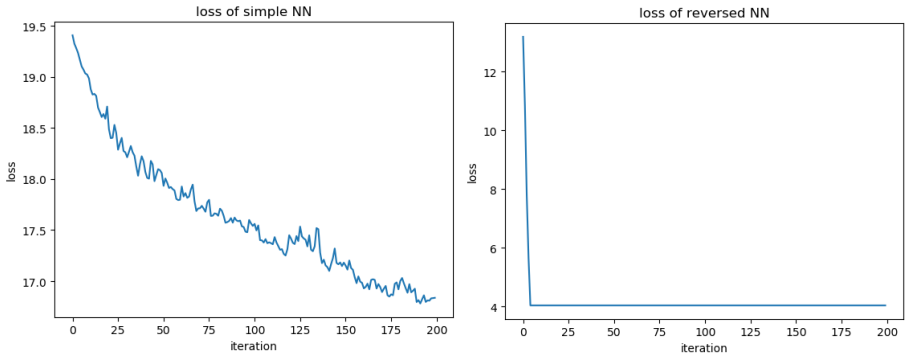

Figure 5 loss updates for BDNN visualization (left: forward network, right: reverse network)

Iterations	Validation accuracy (%)
100	15.92
200	13.78

Table 3 validation accuracy for BDNN w.r.t iterations

We could find that the new model transferred from ResNet18 performs better than BDNN under all circumstances. The BDNN in general is not very suitable for this task since it is sensible to outliers and noises inherent in the SFEW as shown in the previous work.

In the SFEW paper [3], the performance obtained by the pyramid of histogram of oriented gradients (PHOG) is 46.28%. we could notice that the transferred model got a similar performance (43.24%) as in the paper, with a bunch of pre-processing methods assisting. The reason behind this gap may be that in the image detection block of this model, the block would treat the images that failed to detect faces as noises and put it back into the dataset. This to some extent shrinks the original SFEW. The number of iterations may also not be enough for the model to further fine-tuning the parameters.

4. Conclusion and future work

By experimenting on different extents of transferring from ResNet18, a relatively good model for predicting on SFEW was obtained, which consists of a pre-processing block for SFEW and 2 blocks transferred ResNet18. The performance of the model is in all cases better than BDNN and similar to what was obtained in [3]. There are still limitations for this experiment: the resources for training are limited so it is difficult to conduct evaluations on the running efficiency of the transferred model. Also, the SFEW itself is a relatively small dataset, which may make the training phase harder to learning without memorizing. To get some more insight into SFEW as well as transfer learning, future work would focus on the data augmentation to enlarge the size of the dataset for a better generalization. The face detection block of the model may also be transferred for a better capability for SFEW specifically. transferring on some other more complicated models such as ResNet50, Inception-v3, etc. may also be helpful for better performance on predicting SFEW.

Reference

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

[2] A.F. Nejad, T.D. Gedeon: BiDirectional Neural Networks and Class Prototypes, The University of New South Wales, Sydney

[3] Dhall, A., Goecke, R., Lucey, S., & Gedeon, T. (2011, November). Static facial expressions in tough conditions: Data, evaluation protocol and benchmark. In 1st IEEE International Workshop on Benchmarking Facial Image Analysis Technologies BeFIT, ICCV2011.

[4] Smith FW, Rossit S (2018) Identifying and detecting facial expressions of emotion in peripheral vision. PLoS ONE 13(5): e0197160. https://doi.org/10.1371/journal.pone.0197160

[5] Zhiding Yu and Cha Zhang. 2015. Image based Static Facial Expression Recognition with Multiple Deep Network Learning. In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction (ICMI '15).

Association for Computing Machinery, New York, NY, USA, 435–442. DOI:https://doi.org/10.1145/2818346.2830595

[6] Z. Zeng, M. Pantic, G. Roisman, T. S. Huang, et al. Asurvey of affect recognition methods: Audio, visual,and spontaneous expressions. Pattern Analysis andMachine Intelligence, IEEE Transactions on,31(1):39–58, 2009

[7] 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.

[8] Cuda-convnet Google code home page.https://code.google.com/p/cuda-convnet/

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton.Imagenet classification with deep convolutional neuralnetworks. In Advances in neural informationprocessing systems, pages 1097–1105, 2012

[10] O. Russakovsky, J. Deng, H. Su, J. Krause,S. Satheesh, S. Ma, Z. Huang, A. Karpathy,A. Khosla, M. Bernstein, et al. Imagenet large scalevisual recognition challenge. International Journal ofComputer Vision, pages 1–42, 2014

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. arXiv:1409.0575, 2014

[12] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. 2007. Boosting for transfer learning. In Proceedings of the 24th international conference on Machine learning (ICML '07). Association for Computing Machinery, New York, NY, USA, 193–200. DOI:https://doi.org/10.1145/1273496.1273521

[13] Taylor, M.E. and Stone, P., 2009. Transfer learning for reinforcement learning domains: A survey. Journal of Machine Learning Research, 10(Jul), pp.1633-1685.

[14] Zhang, Kaipeng et al. "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks." IEEE Signal Processing Letters 23.10 (2016): 1499–1503. Crossref. Web.

[15] Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.