Facial Expression Recognition with CNN Ensemble

Kuang Liu[1], Minming Zhang[2]
School of Computing Science
Zhejiang University
{liukuang,zhangmm95}@zju.edu.cn

Zhigeng Pan[3]
Institute of Service Engineering
Hangzhou Normal University
zgpan@cad.zju.edu.cn

***Abstract*—This paper is focusing on the Facial Expression Recognition (FER) problem from a single face image. Inspired by the advances Convolutional Neural Networks (CNNs) have achieved in image recognition and classification, we propose a CNN-based approach to address this problem. Our model consists of several different structured subnets. Each subnet is a compact CNN model trained separately. The whole network is structured by assembling these subnets together. We trained and evaluated our model on the FER2013 dataset[7]. The best single subnet achieved 62.44% accuracy and the whole model scored 65.03% accuracy, which is ranked 9th and 5th respectively among all other participants.**

***Keywords*—Facial Expression Recognition, CNN, image classification**

I. Introduction

Facial expression is the most natural way of inner world revelation. It plays a vital role in our social interactions. With it, we can express our feelings, and infer other people's attitude and intention. We are capable of telling other people's facial expressions at a glance. Facial Expression Recognition (FER) system enables machines to infer our intentions through reading the facial expression, that is of great help for Human-Computer Interaction (HCI).

A typical FER system consists of three stages: 1) First, face detection and localization. 2) Next, extract expression information from located faces. 3) Finally, a classifier(like an SVM) is trained on the extracted information to output the final expression labels.

One thing to notice is that many features and algorithms designed for face verification can also be applied to FER. Like Eigenface with principal component analysis (PCA)[2, 15, 5], local binary pattern (LBP) [8, 17, 16], linear discriminate analysis (LDA)[13], independent component analysis (ICA)[3]. And wavelet features including multi-orientation multi-resolution Gabor features[1, 11].

Convolutional Neural Network[12] is a biologically inspired end-to-end model which combines feature extraction and classification together. We can input the raw data, and get the final classified labels without any auxiliary process. In the past five years, mainly due to the developments of "deep learning", CNN has become the most widely used approach in computer vision. It has been demonstrated as an effective model for numerous vision tasks including face detection[9] and verification[20], pose estimation[21], image classification[10, 18, 19], and video tracking[22]. For FER, [14] used a CNN with 2 convolutional layers. [4] tried a much deeper CNN architecture of total 12 layers with 5 convolutional layers. [6] applied CNN to detect smile, and reported significant accuracy improvement.

CNNs have been established as a powerful class of models for image detection, recognition and classification problems. CNNs with tens of millions of parameters can handle massive training samples, and the "features" learned from the networks are all automatic, no handcrafted features needed at all. So CNN can be treated as a powerful automatic feature extractor. Encouraged by the these results CNNs have achieved, we apply the networks to FER problems and evaluate the performance.

We provide a CNN-based approach to address the FER problem in the following three steps: (i) first we describe several different structured CNNs, these subnets are trained separately on the training set; (ii) then these trained CNNs are assembled together by removing the output layers and concatenating the last but one layers together; (iii) lastly, the connected model is trained again to output the final expression labels.

Our contributions can be summarized as follows:

- We propose a CNN-based approach to address the FER problem and empirically evaluate its performance.
- Compared to one single CNN, our network architecture by combining and averaging the outputs of different structured CNNs reports better performance.

II. Proposed Model

The power of CNN models rely on the architecture design with deeper and deeper layers and more and more neurons. But training a big CNN isn't easy. Complexly designed network tend to over-fit the training set, it also needs a huge amount of data to feed in. And training such a big network requires more computing power. In this work, we seek a compact and efficient network design. It should be sufficient to fulfill the task, and easy to train. Inspired by the neat structure VGGNet [18] provides, we design 3 different structured subnets which contain 3 to 5 convolution layers respectively. We denote $subnet_i, i = [1, 2, 3]$ as these 3 Subnets.

A. Subnet architecture

The 3 subnets are outlined in Table I. The only difference among the 3 subnets is the number of convolutional layers. The convolutional layer works like image filters, and is aimed at learning different features. The more convolutional layers, the more detailed features this subnet could learn.

978-1-5090-2303-5/16 $31.00 © 2016 IEEE
DOI 10.1109/CW.2016.34

a) Input Layer.: The raw images which are only pre-processed by subtracting the mean value, are directly passed into the network through the $input$ layer.

b) Convolution Layer.: The kernel size we choose for the convolution layers throughout all 3 subnets is 3×3, which is considered the smallest kernel size to capture the surrounding information. We pad the input image with 1 pixel, and the convolution stride is fixed to 1, so the resolution of the image could be maintained after each convolution.

c) Non-linearity Layer.: Non-linearity layer is followed after every convolution layer. Sigmoid ($f(x) = (1 + e^{-x})^{-1}$), Tanh ($f(x) = tanh(x)$), and Rectified Linear Units (ReLUs, $f(x) = max(0, x)$) are the three most commonly used activation functions. Demonstrated in [10], ReLUs are tend to be several times faster than their equivalents in training. The main advantage of using ReLUs is it can alleviate the gradient vanish problem which is very common in using other two activation functions. Nevertheless, when $x \leq 0$, the ReLU function doesn't active, it filters out the negative responses and obliterate some gradient information.

d) Pooling Layer.: Our network adopts a max pooling strategy with a 2×2 window. The pooling stride is fixed to 2. So there will be no overlapping pooling.

e) Fully Connected Layer.: Fully connected layer can also be treated as convolution layer with a 1×1 kernel size. We stack three fully connected layers at the end of the network.

f) Output Layer.: Softmax is the most commonly used loss function for CNN:

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right) \text{ or } L_i = -f_{y_i} + \log\sum_j e^{f_j}$$

But in this work we choose softmax loss as the output layer loss function:

$$L_i = \sum_{j \neq y_i} \max(0, f(x_i, W)_j - f(x_i, W)_{y_i} + \Delta)$$

For training example (x_i, y_i), the output score is $f(x_i, W)$.The score of the i-th example being the j-th class is $f(x_i, W)_j$.

B. Overall architecture

After describing the previous 3 subnets, we are now ready to build the overall architecture for our CNN. As depicted in Figure 1, our model consists of two stages: (1) The first stage takes a face image as input, and feeds it to the 3 CNN subnets. The 3 subnets contain 8 to 10 layers respectively which is compactly designed, and easy to train. They are the core components of the proposed architecture. (2) The second stage is responsible for predicting the expressions based on the previous stage output. The features extracted by these subnets are concatenated together by adding a fully connected layer at the end. Finally a softmax layer is used as the output layer of the whole network.

With this architecture, we map the face image to one of the seven basic expression labels. We combine the results of different structured CNN models, making them become part of the whole network. Averaging different decisions to get better performance also makes intuitive sense to us, as each CNN subnet could make some error, and they are complementary in working together.

Fig. 1: Overall architecture. The three blue rectangles denote three CNN subnets. Our model is given an face image as input. The results of three subnets are concatenated together in a fully connected layer. The final output layer is the softmax layer.

C. Training Details

a) Data preprocessing and augmentation.: In this work, we resize the face images to 48×48 pixels for analysis. Although nowadays the photo resolution is much higher than this, and decreasing the image resolution actually makes them more difficult to recognize. But low resolution speeds up the training process dramatically without too much impact on the accuracy. For data preprocessing, only zero-mean normalization is needed here. For augmentation, we randomly crop the central regions of the original images, and flip the cropped images horizontally. The augmentation work is considered to reduce the risk of overfitting by increasing the diversity of the training samples.

b) Training.: Mini-batch Gradient Descent is used to train the subnets and the overall model. The batch size is fixed to 100; momentum of 0.9. The weights of network are initialized with small numbers from a normal distribution. The learning rate is set to 0.01 and gradually decrease to 0.001 by hand whenever the validation error stops dropping. The number of epochs varies between 20 to 100 until the loss drops to a reasonable level.

III. Dataset

To evaluate the effectiveness of our method, we use the newly released Facial Expression Recognition 2013 (FER-2013) dataset. The dataset was created using Google image search API with emotion-related keywords. It consists of 48×48 pixel grayscale images of faces. The details of the dataset is shown in Table II. One thing to notice is that the human accuracy on this dataset is $65 \pm 5\%$[7].

IV. Experimental Results

We divided the FER2013 dataset into an 80% training set and 20% validation set. To evaluate our model, we use three different performance metrics: 1) the objective function; 2) the Top1 accuracy of both training and test set; 3) the Top2 accuracy of both training and test set.

TABLE I: Subnet Configurations

$subnet1$	$subnet2$	$subnet3$
input	input	input
conv3-64	conv3-64	conv3-64
maxpool	maxpool	maxpool
conv3-128	conv3-128	conv3-128 conv3-128
maxpool	maxpool	maxpool
conv3-256	conv3-256 conv3-256	conv3-256 conv3-256
maxpool	maxpool	maxpool
FC4096	FC4096	FC4096
FC4096	FC4096	FC4096
FC7	FC7	FC7
SoftmaxLoss	SoftmaxLoss	SoftmaxLoss

TABLE II: The FER-2013 Dataset.

Exp.	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral	Total
Label	0	1	2	3	4	5	6	-
Count	4953	547	5121	8989	6077	4002	6198	35887

The three subnets are trained separately. As we can see in Figure 1, the three subnets are sharing a similar pattern. We repeat the training for 45 to 100 epochs, and they both overfit the training set in the final stages. The results are summarized in Table III.

TABLE III: Subnet Accuracy

ConvNet Config.	top-1 val. acc (%)	top-2 val. acc (%)
$subnet_1$	61.74%	78.72%
$subnet_2$	61.58%	78.83%
$subnet_3$	**62.44%**	**79.80%**

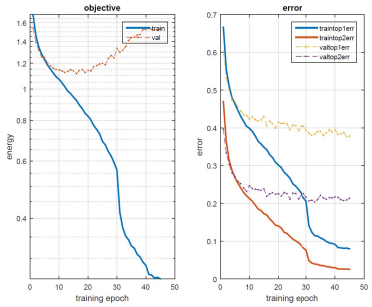

Fig. 2: $subnet_1$ training procedure

The result of each expression of the whole model is shown in Table IV. Note we got very high ($80 \pm \%$) accuracy in "Happiness" and "Surprise". These two expressions are also the easiest two for human to recognize. The overall accuracy of our model on the FER-2013 dataset is 65.03%, which is considered very close to human level $65 \pm 5\%$.

Table V shows the leaderboard of facial expression recognition challenge[1] on FER2013 dataset. As you can see, our best single model achieved the accuracy of 65.03%, which is ranked #9, and finally the ensembled model ranked #5 among all the participating teams.

[1]https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/

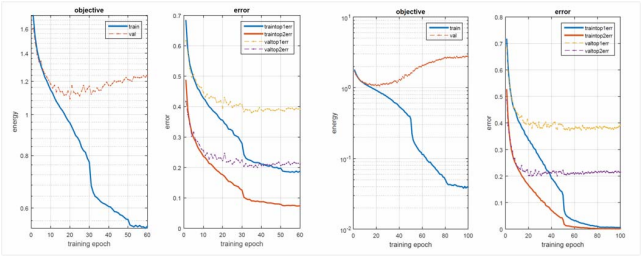

Fig. 3: training precedure of $subnet_2$ (left) and $subnet_3$ (right)

TABLE IV: The Whole Network Accuracy.

Expression	Precision	Recall	F-Score	Accuracy
Angry	0.6017	0.7172	0.6544	62.12%
Disgust	0.7073	0.5918	0.6444	67.35%
Fear	0.5859	0.6536	0.6179	59.59%
Happy	0.8945	0.6794	0.7722	79.96%
Sad	0.6107	0.4580	0.6235	58.30%
Surprise	**0.9275**	0.6769	**0.7826**	**81.20%**
Neutral	0.8192	**0.8129**	0.6976	64.90%

V. Conclusion

In this paper, we address the FER problem with a CNN ensemble model. We design three different structured CNN subnets. An overall architecture is introduced later to incorporate the output of these subnets. Furthermore, we train and evaluate the performance of our model on the FER2013 dataset. The main advantage of the proposed model is we focus on different CNNs rather than one. It's easy to get better performance by combing all the results together.

In future work, we would like to incorporate human engineering features (like LBP) into our model. And modify the model structure to explore how it performs for different vision tasks.

VI. Acknowledgement

This work is supported by Project NSFC (Grant No. 61332017, 61572243).

TABLE V: FER-2013 Leaderboard

RANK	TEAM	ACC. (%)
1	RBM	69.77
2	UNSUPERVISED TEAM	69.07
3	MAXIM MILAKOV	68.15
4	RADU+MARIUS+CRISTI	67.48
-	**Subnet Ensemble**	**65.03**
5	LOR.VOLDY	64.56
⋮		
8	XAVIER BOUTHILLIER	62.78
-	**Subnet3**	**62.44**
9	SAYIT	61.91
10	ALEJANDRO DUBROVSKY	61.38
⋮		
56	DSTARERSTOR	20.40

REFERENCES

[1] *3rd International Conference on Face & Gesture Recognition (FG '98), April 14-16, 1998, Nara, Japan*. IEEE Computer Society.

[2] Nikunj Bajaj, Aurobinda Routray, and S. L. Happy. Dynamic model of facial expression recognition based on eigen-face approach. *CoRR*, abs/1311.6007, 2013.

[3] I. Buciu, C. Kotropoulos, and I. Pitas. Ica and gabor representation for facial expression recognition. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, pages II – 855–8, 2003.

[4] P. Burkert, F. Trier, M. Zeshan Afzal, A. Dengel, and M. Liwicki. DeXpression: Deep Convolutional Neural Network for Expression Recognition. *ArXiv e-prints*, September 2015.

[5] Andrew J Calder, A. Mike Burton, Paul Miller, Andrew W Young, and Shigeru Akamatsu. A principal component analysis of facial expressions. *Vision Research*, 41(9):1179–1208, 2001.

[6] Patrick O. Glauner. Deep convolutional neural networks for smile recognition. *CoRR*, abs/1508.06535, 2015.

[7] Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, and Dong Hyun Lee. Challenges in representation learning: A report on three machine learning contests. *Neural Networks*, 64:117–124, 2014.

[8] S. L. Happy and Aurobinda Routray. Robust facial expression classification using shape and appearance features. In *Advances in Pattern Recognition (ICAPR), 2015 Eighth International Conference on*, pages 1 – 5, 2015.

[9] Ilya Kalinowski and Vladimir Spitsyn. Compact convolutional neural network cascade for face detection. *CoRR*, abs/1508.01292, 2015.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[11] Weifeng Liu and Zengfu Wang. Facial expression recognition based on fusion of multiple gabor features. In *Pattern Recognition, International Conference on*, pages 536–539, 2006.

[12] Yann Lécun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] P. Marasamy and S. Sumathi. Automatic recognition and analysis of human faces and facial expression by lda using wavelet transform. In *Computer Communication and Informatics (ICCCI), 2012 International Conference on*, pages 1 – 4, 2012.

[14] A. Mollahosseini, D. Chan, and M. H. Mahoor. Going Deeper in Facial Expression Recognition using Deep Neural Networks. *ArXiv e-prints*, November 2015.

[15] Sohini Roychowdhury. Facial expression detection using patch-based eigen-face isomap networks. *CoRR*, abs/1511.03363, 2015.

[16] Caifeng Shan, Shaogang Gong, and Peter W. McOwan. Robust facial expression recognition using local binary patterns. In *ICIP*, pages 370–373. IEEE, 2005.

[17] Caifeng Shan, Shaogang Gong, and Peter W. Mcowan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image & Vision Computing*, 27(6):803–816, 2009.

[18] Karen Simonyan, Andrew Zisserman, Karen Simonyan, and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Eprint Arxiv*, 2014.

[19] C. Szegedy, Wei Liu, Yangqing Jia, and P. Sermanet. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[20] Y. Taigman, Ming Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708, 2014.

[21] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1653–1660, 2014.

[22] Li Wang, Ting Liu, Gang Wang, Kap Luk Chan, and Qingxiong Yang. Video tracking using learned hierarchical features. *Image Processing IEEE Transactions on*, 24(4):1424–1435, 2015.