Multi-Objective Convolutional Learning for Face Labeling

Xianan Huang Yan Zhao
Department of Mechanical Engineering
{xnhuang,zhaoyann}@umich.edu

Tianyu Jiang Yunfan Liu
Department of Electrical Engineering and Computer Science
{jiangty, yunfan}@umich.edu

Abstract

In this project report, we present our investigation on the paper "Multi-Objective Convolutional learning for Face Labeling" proposed by Liu *et al.* [1]. Their work introduces a novel multi-objective learning method that optimizes a unified convolutional neural networks (CNNs) with two distinct loss-functions which encode the label likelihoods and dependencies respectively. In addition to the RGB input image, a non-parametric prior is employed as a new input channel to regularize the network. State-of-the-art performance has been achieved on challenging benchmark dataset LFW and Helen using the proposed algorithm. In this report we first review the theoretical basis related to convolutional neural network and then we introduce the algorithm proposed by Liu *et al.* in detail. We also conducted experiments to reproduce the baseline evaluation in the paper, and result analysis is provided to show we have achieved basic facial components labeling and the performance could be improved by larger and deeper network configuration.

1 Introduction

Convolutional neural networks (CNNs) has long been studied since first introduced in 1998 [2]. As powerful non-linear classification models, CNNs are capable of generating adaptive and discriminative features compared to the traditional classifiers which require features extracted in advance. Liu *et al.* [1] proposed a novel multi-objective learning method to reduce the computational cost and avoid overfitting by decomposing the CRF loss into a unary term that employs softmax loss and a pairwise term using logistic loss. An unified weight-sharing network is optimized to minimize of the two losses together. The proposed method could be trained as efficiently as other CNNs, and region boundaries could be learned by converting logistic loss into a 'edge vs. non-edge' classification problem.

Face labeling is the problem in which every pixel of the image is assigned a label of facial component so that the facial image is segmented into distinct regions. Various algorithms have been developed based on landmarks along face contour and components. These methods suffer from pose, illumination variations and resulting occlusions. However, CNNs provide a way to extract robust representations of facial images. The proposed algorithm is applied to face labeling problems and state-of-the-art results are obtained on challenging benchmark datasets. Considering faces are highly structures patterns, a non-parametric prior is employed as an additional input channel. Experiment results indicate that by adding the non-parametric prior the scale of the network is significantly reduced while maintaining the competitive performance.

2 Related Work

Combination of CNNs and graphical models has been studied in several recent works. Multi-scale CNNs and region tree structure are combine to solve scene parsing problem in the work of Farabet *et al* [3]. However, they train the CNN and the graphical model in a sequential manner while in this work the CRF and CNN are optimized jointly.

Joint training of CNNs and graphical models also have been studied in [4, 5]. However, these methods differ from Liu *et al*'s work in that they either do not achieve weight sharing [4], resulting in expensive computational cost, or require two step training [5]. The novelty of [1] lies in joint training process of CRF and CNN with weight sharing through multi-objective optimization, yet the non-parametric prior has not been applied in the above related work.

Applications of CRF on face labeling problem have been reported in recent works [6, 7]. In the work of Warrell and Prince [6] , the facial structure is modeled using a family of multinomial priors and a CRF is employed to labeling facial components. In [7], the face shape prior is modeled using a restricted Boltzmann machine and combined with CRF for face labeling. Unlike [1] which uses CNN as an end-to-end classifier, these two methods are based on hand-extracted features. A hierarchical face parsing system is proposed by Luo *et al*, while in Liu *et al*'s work the model is trained in a single pipeline.

3 Methodologies

In order to present a systematic introduction of the algorithm in the work of Liu *et al.* [1], we first review the theoretical basis of CNNs and CRF models. Based on that, we explain the multi-objective convolutional learning method proposed in detail, including unary and pairwise term decomposing and non-parametric prior.

3.1 Convolutional neural networks (CNNs)

Convolutional neural networks consist of convolutional layers and fully connected layers, which can be regarded as feature extraction and classification tool respectively. A convolutional layer are usually followed by several post processing layers such as pooling layer, normalization layer, and rectified linear unit. A convolutional layer consists of several filters whose weights and biases are usually of the same size, and the input is convoluted with each of these filters to generate a feature map. The pooling layers are all designed to select the maximum value of every non-overlapping areas so as to reduce the effect of small translations in image features. The normalization layers applied in this paper use local region normalization algorithm to accelerate training process by reducing internal covariant shift. The rectified linear unit is used as the activation function for effective gradient back propagation. Fully connect layers are added to the end of the convolutional layers with dropout layers added to the end of every linear layers to prevent overfitting problem.

During training process, all the parameters, including the convolution filters, weighting parameters and bias parameters in both convolutional and fully connected layers, are randomly initialized. Losses for different objectives are calculated in inference phase. Then, all parameters are updated with stochastic gradient descent method in back propagation phase. Thus, the CNN is expected to catch discriminative features and fully connected layer are expected to classify based on the features fed iteratively and automatically.

3.2 Conditional random fields (CRFs)

CRFs are a family of statistical modeling methods used for structured prediction which superior to ordinary classifiers in that context information is taken into account [10]. In a face modeling problem, the image is modeled as a undirected graph $\mathbf{G} = (\mathcal{V}, \mathcal{E})$ defined on an image where each pixel is treated as a node in $\mathcal{V}$ and $\mathcal{E}$ is a 4-neighbors connectivity. The CRF model could be formulated as

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z(\mathbf{X})}\text{exp}[-E(\mathbf{Y}, \mathbf{X})] \tag{1}$$

where $\mathbf{X}$ is the input image, $\mathbf{Y}$ a set of random variables representing labels, and $\mathbf{Z}$ a partition function on $\mathbf{X}$. The energy function $E(\mathbf{Y}, \mathbf{X})$ consists of two independent terms

$$E(\mathbf{Y}, \mathbf{X}) = \sum_{i \in \mathcal{V}} E_u(y_i, \mathbf{x}_i) + \lambda \sum_{(i,j) \in \mathcal{V}} E_u(y_i, y_j, \mathbf{x}_{i,j}) \tag{2}$$

In (2), $E_u(y_i, \mathbf{x}_i)$ is the unary term measuring the cost of assigning a certain label y_i, and a multi-class classifier is introduced to model the label assignment cost of unary term $E_u(y_i, \mathbf{x}_i; w_u) = -\text{log}P_u(y_i = l|\mathbf{x}_i, w_u)$. A new label z_{ij} is introduced in order to measure the similarity of adjacent variables y_i, y_j

$$z_{ij} = \begin{cases} 1 & \text{if } y_i = y_j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Thus the classifier for pairwise term is reduced to a binary one, that is $E_p(y_i, y_j, \mathbf{x}_i; w_p) = -\text{log}P_p(z_{ij} = 1|\mathbf{x}_{ij}, w_p)$. In Liu *et al*'s work, a single unified CNN is trained aiming at sharing weights and features during the optimization process, so that the computational complexity and the chance of overfitting are greatly reduced.

Two distinct loss functions are defined for unary and pairwise terms respectively. By defining the output of the top layer of CNN (extracted feature) as $h_i = h(\mathbf{x}_i, w)$, the unary classifier could be expressed using a softmax function

$$P_u(y_i = l|h_i, w_u) = \frac{\text{exp}(w_u^l \cdot h_i)}{\sum_{l=1}^{K} \text{exp}(w_u^l \cdot h_i)} \tag{4}$$

and the corresponding softmax loss is $L(y_i, \mathbf{x}_i, w, w_u) = -\text{log}P_u(y_i = l|h_i, w_u)$. The pairwise term is given by a logistic function

$$P_p(z_{ij} = 1|h_i, w_p) = \frac{1}{1 + \text{exp}(-w_u^l \cdot h_i)} \tag{5}$$

and the loss of the pairwise term is also given by $L(z_{ij}, \mathbf{x}_{ij}, w, w_p) = -\text{log}P_p(z_{ij} = 1|h_i, w_p)$

Given the two loss functions, a unified CNN is trained through multi-objective optimization by updating gradients of both softmax and logistic loss functions for back-propagation. The optimization problem could be expressed as $w = \text{min}_w\{O_u(w, w_u), O_p(w, w_p)\}$ where the expected losses could be expressed as ($\Psi(w, w_u)$ and $\Phi(w, w_p)$ are regularization terms)

$$\begin{cases} O_u(w, w_u) = \mathbb{E}(\sum_{i \in \mathcal{V}} L_u(y_i, \mathbf{x}_i, w, w_u)) + \Psi(w, w_u) \\ O_p(w, w_p) = \mathbb{E}(\sum_{i,j \in \mathcal{E}} L_p(z_{ij}, \mathbf{x}_{ij}, w, w_p)) + \Phi(w, w_p) \end{cases} \tag{6}$$

There are mainly three advantages of the proposed model: First, both of the two loss functions make use of the same features generated by a unified CNN. Second, the back-propagation process involves the error comes from both objectives based on which the network weights could be learned. Third, training efficiency is guaranteed and and overfitting problem is avoided by using a unified network.

3.3 Non-parametric prior

In order to obtain more information from the image dataset, a nonparametric prior is constructed for every image in the testing and training set according to Liu *et al.* [1]. A pre-trained CNN [8] is used to detect five facial landmarks (nose, eyes and mouth corners) for all the images. After that principal component analysis (PCA) is performed on the ground truth of images in validation set and the principle components are used to construct a shape subspace. Given a subject image from the training/testing set, 10 exemplars from the validation set with the smallest Euclidean distances of PCA coefficients are selected. Then each of the 10 exemplar images is aligned with the subject image using a similarity transformation estimated by the corresponding facial landmarks. The non-parametric prior is the average of the aligned exemplar images. The prior could be understand as the possibility of each label value at every pixel of subject image.

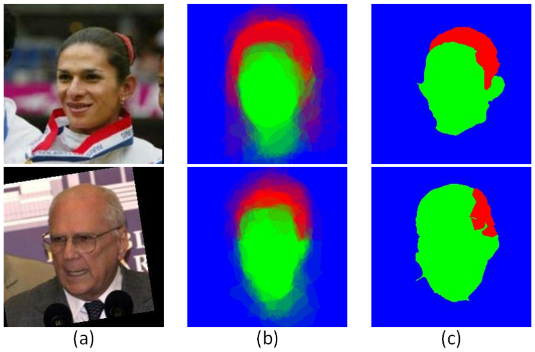

Figure 1: Examples of non-parametric. (a) subject images (b) prior image (c) ground truth

4 Experiment

4.1 LFW-PL dataset and patch selection

For fair comparison reasons, we use the same LFW-PL dataset as Liu *et al* do in the paper. LFW-PL stands for LFW *part labels dataset*, which is a public-available subset of the original benchmark dataset LFW [7]. LFW-PL dataset contains 2927 images of size 250 × 250 with manually annotated labels (skin, hair and background). It is divided into a training set with 1500 images, a testing set of size 927 and a validation set used for prior computation with 500 images. However, facial landmarks of part of the images cannot be detected using the above method[8]. Considering the small portion(154 out of 2927) of the images cannot be marked, we simply discard them and make use of the rest valid images for prior computation, training and testing. We randomly select 600 patches per image from the training images and 30000 patches in total from the testing images as testing set.

4.2 Reproducing experiments of Liu *et al.*

4.2.1 CNN training without prior

We first conducted experiment without non-parametric experiment. We employed the same CNN architecture as Liu *et al.* proposed. The inputs are 72 × 72 single scale patches sampled from images of training set. The first two layers of CNN are convolutional layers with filters of size 5 × 5 followed by a max pooling with a downsampling stride of 2. After that are 5 consecutive convolutional layers with filter size of 3 × 3 without pooling layers. All the convolutional layers are followed by rectification (ReLu) non-linearity unit and a local response normalization(LRN) layer. On top of the convolutional layers, a fully connected network is trained as a classifier with 3 output units, corresponding to 3 labeling classes. An overview of the system is shown in 2

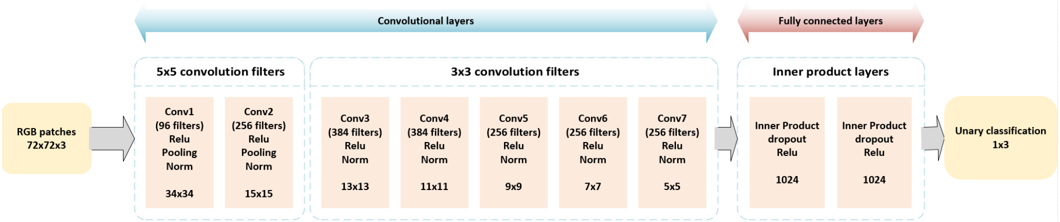
Convolutional layers
Fully connected layers
5x5 convolution filters
3x3 convolution filters
Inner product layers
RGB patches 72x72x3
Conv1 (96 filters) Relu Pooling Norm 34x34
Conv2 (256 filters) Relu Pooling Norm 15x15
Conv3 (384 filters) Relu Norm 13x13
Conv4 (384 filters) Relu Norm 11x11
Conv5 (256 filters) Relu Norm 9x9
Conv6 (256 filters) Relu Norm 7x7
Conv7 (256 filters) Relu Norm 5x5
Inner Product dropout Relu 1024
Inner Product dropout Relu 1024
Unary classification 1x3

Figure 2: Author proposed CNN structure

Then Caffe [9], a deep learning programming framework, was utilized, aiming at exploiting the computation capacity of GPU to do the job. The training process is carried out using mini-batch stochastic gradient descent with momentum, weight decay, dropout ratio and batchsize to be 0.9, 5×10^{-4}, 0.5 and 50 respectively, with are same as the configuration in [1]. Learning rate is manually decreased by a factor of 10 when the loss is observed fluctuating. The input data is converted into .lmdb format for efficiency consideration.

It took more than 8 hours to do the training process on NVIDIA GeForce GTX 770M of all 30000 iterations. The experiment result shows that the loss of the objective function fluctuates greatly and does not decrease obviously. Inappropriate parameters and inadequate iteration times might be the cause of this phenomenon. However, considering the limited computational and time resources we have, the expense of tuning and searching for optimal parameters is unaffordable for us. Therefore, we decided to reduce the scale of the network as is shown in Fig. 3. Note that if there is no prior channel, the actual input size is 72 × 72 × 3.

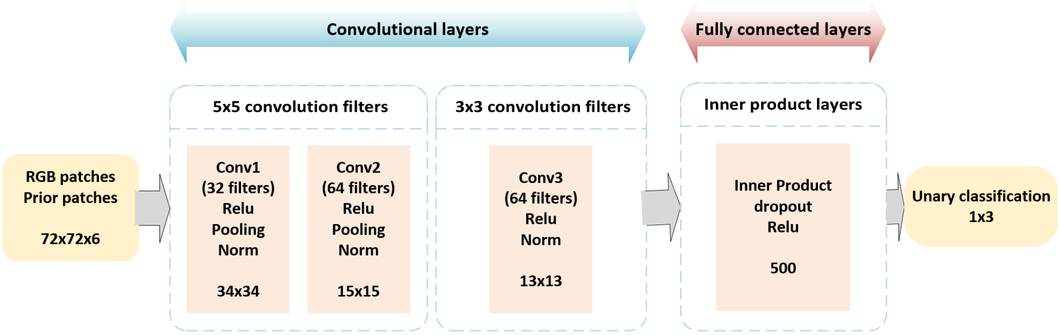

Figure 3: Directly simplified CNN structure

The CNN structure shown in Fig. 3 is a directly reduced version of the original complete CNN. Fewer filters are used in the first two convolutional layers with size of 5, and the following stack of convolutional layers is compacted to a single convolutional layer. The fully connected network is also reduced to a single stage with less neurons. We trained the models with 60000 iterations and the test accuracy with the optimal parameters is 82.8% over all the classes as oppose to 92.92% in [1]. A series of models such as LeNet structure shown in Fig. 4 are also tuned and tested on this application to assess labeling accuracy. However, all of our experiments achieves similar accuracy, ranging from 81% to 83%.

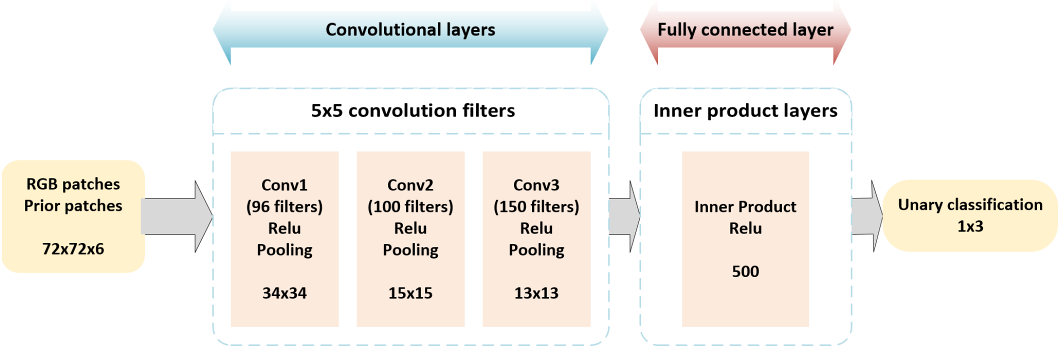

Figure 4: LeNet structure

4.2.2 CNN training with prior input

We make use of the non-parametric prior corresponding to each patch as additional input channel. Therefore the size of the input turns into $72 \times 72 \times 6$, and the input data is normalized and converted into .hdf5 format for computational efficiency. Experiments are conducted on CNNs with structure shown in Fig. 3 and 4. However, again the loss does not decrease obviously and fluctuates greatly. The reason for this phenomenon is that since the input with prior contains much more information than before, a shallow network cannot capture the distinctive features in the input data anymore. The memory and time expense also increases greatly after involving the prior information so that we can either build a deeper network or search for optimal parameters based on the computational resources we have.

5 Evaluation

5.1 Results without Prior

The experiment is conducted with a desktop computer equipped with CPU i7-3930K and GPU Nvidia GTX 560 Ti. It took 3 hours to run 600000 iterations with 100000 input pitches, with final loss fluctuating around 0.3. The test accuracy over 30000 randomly selected patches is 0.828. A comparison of forward inference labeling result and ground truth is shown below in Fig. 5. By visually evaluation of the results, the team notices that the light-colored skin are all classified as face portion including hands and faces in the background. Dark-colored faces are not well classified as shown in Fig. 5. Some portion of the face are misclassified as hair. Also, the dark-colored background are often misclassified as hair. The hypothesis for this result is that with the constraint of memory size, the size of convolution filter is constrained, thus the features captured are not enough for accurate classification. A detailed discussion of influence of convolution filter is presented in the following section.

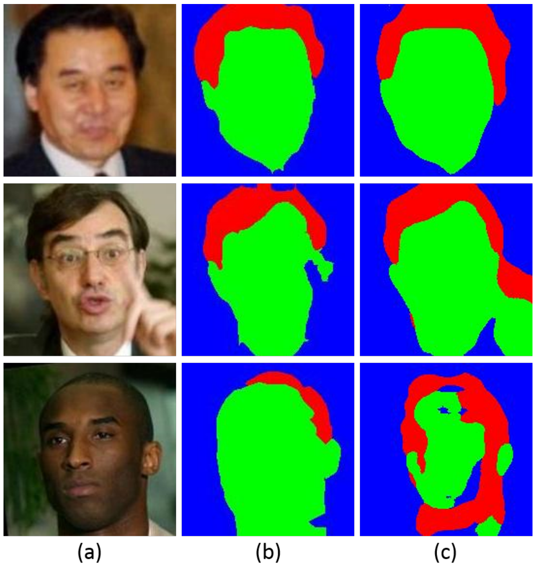

Figure 5: Results of Directed Simplified CNN Structure with (a)Original Picture (b)Ground Truth (c)Forward Inference Labeling Result

5.2 Trained Network

The first convolutional layer is extracting features from the original input patches. Therefore, the first convolutional layer of a well-trained CNN is expected to possess observable pattern, such as 2D edge detector pattern in different directions or certain color filtering pattern, and have visible effect if applied to an image. The first convolutional layer in the exemplar mat provided by author consists of 96 5x5 convolutional filters with their corresponding bias. Considering that 5x5 filter images are not visually explanatory, we examine this layer by apply the filters and their bias to the original image and get 96 filtered images in Fig. 6 For example, filter 1 and 89 select short horizontal edges;

filter 4 and 32 extracts high frequency components such as hair; filter 57 is extracting vertical edges; filter 38 detects face blobs. The effect of those filters are distinguishable and have the same effect on many other images in the set. Higher layer outputs are not as intuitive as the first convolutional layer.

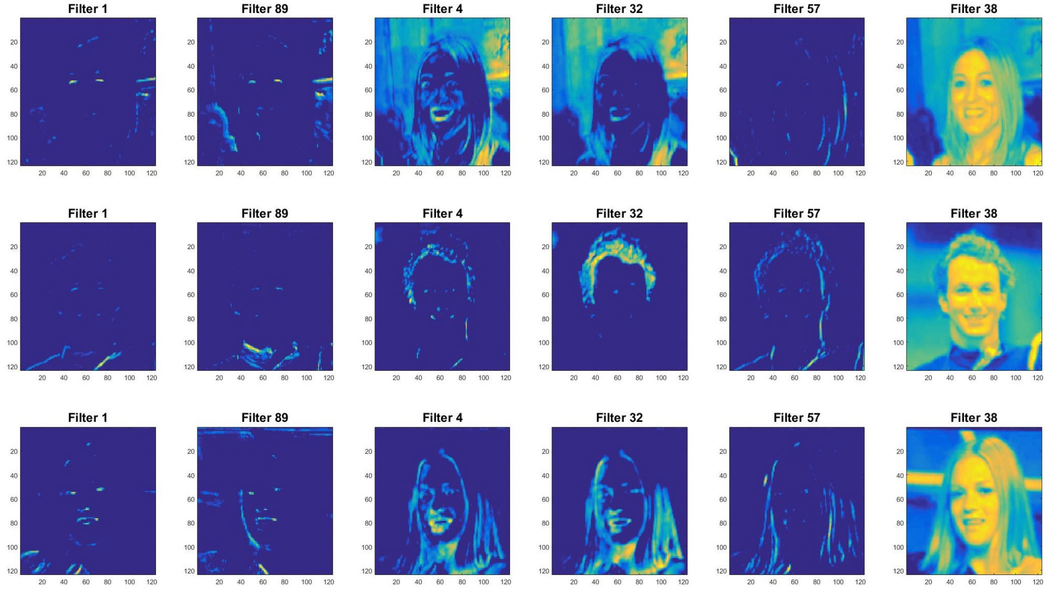

Figure 6: Effect of author's first convolutional layer

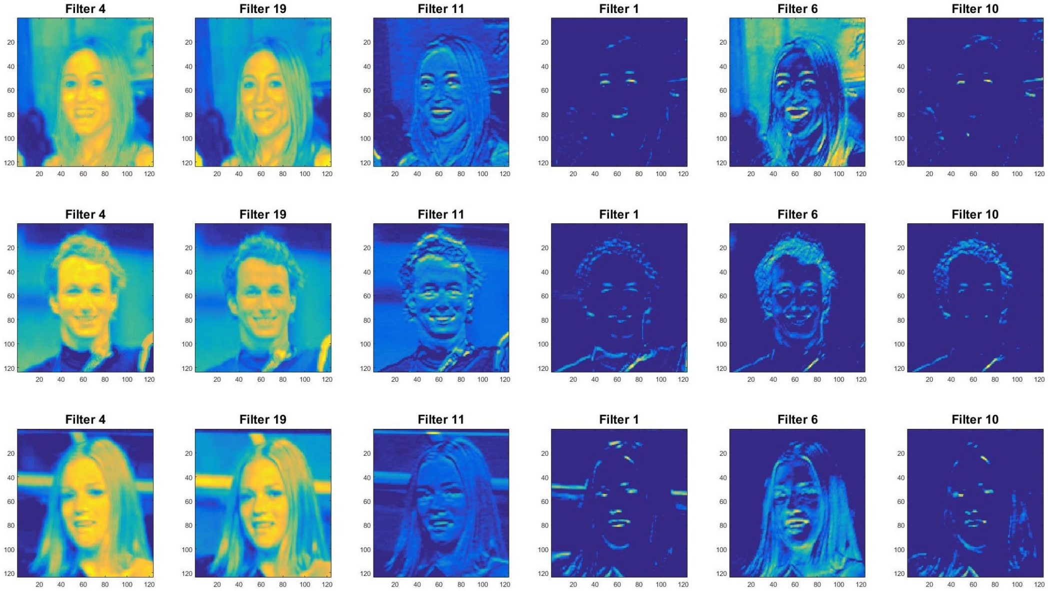

Figure 7: Effect of LeNet's first convolutional layer

Similarly, we applied the LeNet's first convolutional layer of the simplified CNN to images and get Fig. 7. For example, filter 4 and 19 are detects facial blobs; filter 11, 1 and 10 highlights edges; filter 6 selects high frequency components. Same effect occurs on many different images when applying these filters. This indicate that LeNet CNN architecture is trained to extract features as designed but these filters are not as powerful as those from the author's network. Thus, the low classification ability of this CNN architecture is probably due to the lack of filters in every convolutional filter.

The effect of the convolutional layer of the simplified CNN is much more obvious than that of the LeNet CNN as shown in Fig. 8. For example, filter 1and 18 selects different facial features; filter 19 and 27 extracts high frequency components like hair; filter 21 and 23 selects slightly different horizontal edges. Besides, this CNN has more filters than LeNet CNN, the ability of each filter are more observable, and has much better classification balance than that of the LeNet CNN indeed.

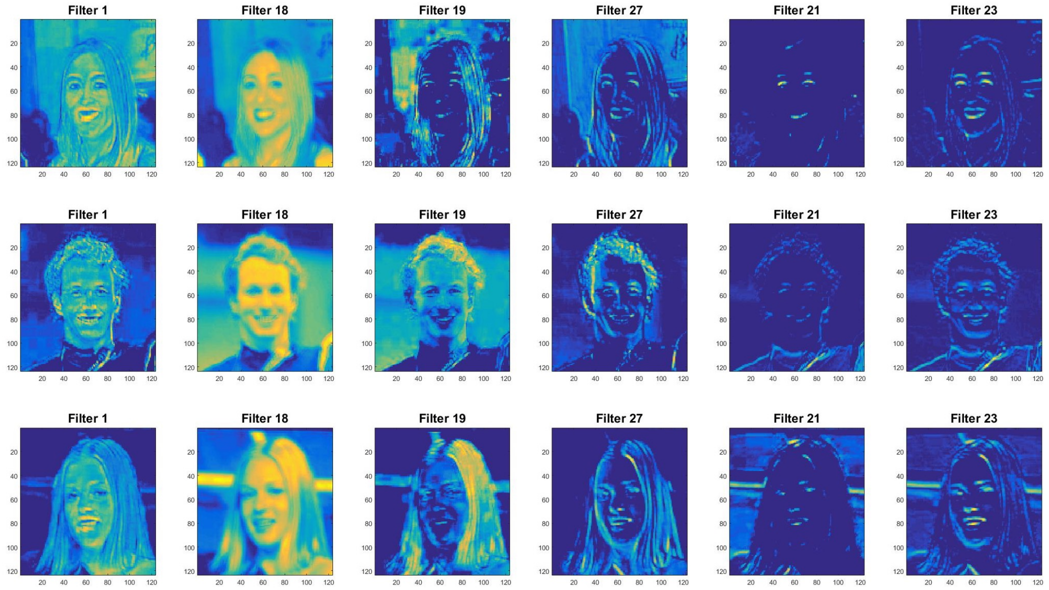

Figure 8: Effect of simplified CNN's first convolutional layer

Besides, the CNNs corresponds to Fig. 7 and Fig. 8 do not include pairwise classification loss. However, many filters show edge detecting ability in these two CNN. This give rise to the fact that edge detection may be well combined with face-hair-and-backgound semantic classification, and even beneficial.

6 Conclusions

The team successfully demonstrated training a simplified CNN structure through Caffe. Results show that with the simplified structure, the performance is not as good as the published network. The parameters in regularization terms and convolution need to be well tuned to get good performance. Besides, the stopping criteria can be tricky due to the non-convex nature of CNN. However, after well-tuned, the performance of CNN is robust with respect to different features.

7 Description of individual effort

Xianan Huang: Prior algorithm implementation, CNN network training with simplified structure, Caffe solver parameter tuning

Tianyu Jiang: Design CNN architecture, convolutional layer analysis.

Yunfan Liu: CNN training on several models and data patterns, different data preparation strategies, report writing

Yan Zhao: Literature review, report writing, IDE configuration

References

[1] Liu, Sifei, et al. ”Multi-Objective Convolutional Learning for Face Labeling.” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[2] LeCun, Yann, et al. ”Gradient-based learning applied to document recognition.” Proceedings of the IEEE 86.11 (1998): 2278-2324.

[3] Farabet, Clement, et al. ”Learning hierarchical features for scene labeling.” Pattern Analysis and Machine Intelligence, IEEE Transactions on 35.8 (2013): 1915-1929.

[4] Ranftl, Ren, and Thomas Pock. ”A deep variational model for image segmentation.” Pattern Recognition. Springer International Publishing, 2014. 107-118.

[5] Tompson, Jonathan J., et al. ”Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation.” Advances in Neural Information Processing Systems. 2014.

[6] Warrell, Jonathan, and Simon JD Prince. ”Labelfaces: Parsing facial features by multiclass labeling with an epitome prior.” Image Processing (ICIP), 2009 16th IEEE International Conference on. IEEE, 2009.

[7] Kae, Andrew, et al. ”Augmenting CRFs with Boltzmann machine shape priors for image labeling.” Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, 2013.

[8] Y. Sun, X. Wang, and X. Tang, ”Deep convolutional network cascade for facial point detection”, *CVPR*, 2013.

[9] Jia, Yangqing, et al. ”Caffe: Convolutional architecture for fast feature embedding.” Proceedings of the ACM International Conference on Multimedia. ACM, 2014.

[10] R. Ranftl and T. Pock. A deep variational model for image segmentation. In X. Jiang, J. Hornegger, and R. Koch, editors, Pattern Recognition, Lecture Notes in Computer Science, pages 107118. Springer International Publishing, 2014. 1, 2