IPSJ Transactions on Computer Vision and Applications Vol.7 99–103 (July 2015)

[DOI: 10.2197/ipsjtcva.7.99]

Express Paper

Cost-alleviative Learning for Deep Convolutional Neural Network-based Facial Part Labeling

Takayoshi Yamashita[1,a)] Takaya Nakamura[1] Hiroshi Fukui[1,b)]
Yuji Yamauchi[1,c)] Hironobu Fujiyoshi[1,d)]

Received: March 13, 2015, Accepted: August 20, 2015, Released: July 27, 2015

Abstract: Facial part labeling which is parsing semantic components enables high-level facial image analysis, and contributes greatly to face recognition, expression recognition, animation, and synthesis. In this paper, we propose a cost-alleviative learning method that uses a weighted cost function to improve the performance of certain classes during facial part labeling. As the conventional cost function handles the error in all classes equally, the error in a class with a slightly biased prior probability tends not to be propagated. The weighted cost function enables the training coefficient for each class to be adjusted. In addition, the boundaries of each class may be recognized after fewer iterations, which will improve the performance. In facial part labeling, the recognition performance of the eye class can be significantly improved using cost-alleviative learning.

Keywords: facial part labeling, cost-alleviative learning, convolutional neural network, cost-function, back propagation

1. Introduction

Facial part labeling involves parsing semantic components of the face such as the mouth, nose, and eyes. This labeling enables high-level facial image analysis, and contributes greatly to face recognition, expression recognition, animation, and synthesis. Facial point detection and face alignment are major components in identifying facial parts. Existing research focuses on detecting landmarks such as eye and mouth corners, and can generally be categorized into discriminative and regression methods [1], [2], [8], [10] or graphical model methods [7], [14]. These approaches tend to refine each landmark from the initially estimated location. Whereas facial point detection and face alignment obtain facial landmarks with high accuracy, post-processing or human interaction is required to label each pixel with the name of that facial part. In addition, it is hard to define how many landmark positions are necessary. To address this problem, semantic segmentation approaches have been proposed [3], [9], [11]. Deep Learning achieves state-of-the-art performance for the labeling of facial parts [9], [11] and scenes [3]. In particular, the Deep Convolutional Neural Network (DCNN) has attracted attention for applications in computer vision tasks [5], [6]. However, DCNN-based methods cannot label eye regions correctly, as shown in **Fig. 1**. The DCNN trains the network parameters by iteratively updating according to the error between prediction and ground truth. If the prior probabilities are biased to a particular class, the error in classes with lower probabilities may not be propagated correctly. In Ref. [3], superpixel-based segmentation was used to reduce the bias and ensure that errors were propagated appropriately. However, the accuracy is significantly affected by the superpixel segmentation. A different approach uses a Deep Belief Network to detect and label all facial part regions at the same time [9]. Although this method avoids the problems of superpixel segmentation by detecting a larger region, it does require a complex architecture. These techniques can achieve good performance, but do not solve the training problem for the labeling of small regions.

In this paper, we consider this problem and propose a training approach to improve the accuracy of the labeling of small regions. As the conventional cost function handles the error in all classes equally, the error in a class with a slightly biased prior probability

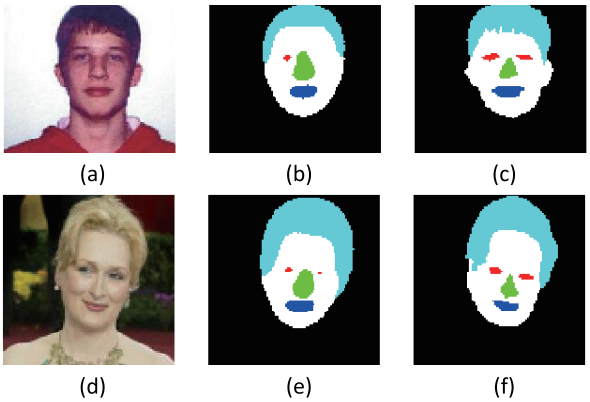

Fig. 1 Facial parts labeling based on conventional DCNN. (a) and (d) are input images, (b) and (e) are the resulting images, (c) and (f) are the ground truth.

[1] Chubu University, Kasugai, Aichi 487–8501, Japan
[a)] yamashita@cs.chubu.ac.jp
[b)] fhiro@vision.cs.chubu.ac.jp
[c)] yuu@vision.cs.chubu.ac.jp
[d)] hf@cs.chubu.ac.jp

tends not to be propagated. We propose a cost-alleviative learning method that weights the cost function using the prior probability of each class. We describe the details of cost-alleviative learning in Section 3. We show that the cost weight can control the accuracy and convergence speed of certain classes, and compare manually set probabilities with those based on the ratio of facial part regions in an image. In Section 4, we demonstrate the superiority of our cost-alleviative learning over the conventional method.

2. Related Work

Conventional applications of facial analysis employ facial point detection or face alignment as preprocessing techniques to normalize the facial region. The Active Shape Model (ASM) [1] is a well-known face alignment approach, and has inspired many methods [10], [13]. ASM depends on the initial position of the model, and does not work well in an unconstrained environment. The Markov Shape Model [7] uses multiple initial shapes as local line segments and appearances. While these methods reduce the influence of initialization, they can be time consuming. Discriminative approaches adapt to pose variations and cluttered backgrounds, and generally offer reduced computational cost [2], [8], [14]. The regression ASM detects small patches related to facial components by boosting the classifiers [2]. As the detection process for each patch is independent, the matching process may fail if there are several missing components.

Representations that use pixel-wise label maps provide richer information and more robustness than facial point detection and face alignment [12], [15]. For example, the scene parsing approach in Ref. [15] constructs a topological structure model of the face image that is robust in unconstrained environments. As the model is based on rough priors, it is inaccurate for small face components such as the eyes. A hierarchical approach to face parsing can improve the accuracy of these small components [9]. First, face parts (e.g., upper face, lower face) are detected, before a second stage detects face components (e.g., mouth, eyes) and a final, third stage segments the components according to pixel-wise labels. This hierarchical structure is constructed by heuristic knowledge, and is not easily generalized. DCNN-based approaches have been successfully applied to scene labeling tasks [3]. This method also employs a hierarchical structure that segments and labels superpixels.

For recognition tasks, DCNN updates the network parameters according to the error obtained from several images. Although particular classes may be biased in each iteration, the update process uses equal prior probabilities for all classes over all iterations. On the other hand, for semantic segmentation, the errors for all pixels are calculated and propagated at the same time. In particular, in the case of facial part labeling, the prior probabilities of each facial part will be different. Whereas a class with a high prior probability is likely to propagate the error to the next step, one with a low probability will find convergence difficult, with a large error that does not occur in the next step. As a result, the performance for this class will be significantly degraded.

In this paper, we propose a cost-alleviative learning technique that introduces a new cost function to improve specific classes in the semantic segmentation task. The cost function is weighted based on the prior probability of each class. These cost weights make it possible to alleviate the propagation of error. We now describe the cost function and the efficiency of the cost weights in detail.

3. Proposed Method

Figure 2 shows the network structure of DCNN for semantic segmentation. The network is built of three types of layer: convolutional, pooling, and a full connection layer. The convolution and pooling layers are arranged successively as a deep structure, and are followed by the full connection layer. The output layer has label maps for each class. In the training phase, the errors are calculated as the difference between the response value and labeled data. The update values are derived from the errors, and are propagated by back-propagation. The error E_c of output unit c is defined in Eq. (1).

$$E_c = -\sum_{p \in P} t_p \log y_p \tag{1}$$

Note that P is the mini-batch size, t_p are the labeled data, and y_p is the prediction result. The total training error on one iteration is computed by adding the errors in all classes. The conventional cost function treats these errors equally. We set the cost weight v_c for each class c, and define the total training error E_M as the weighted sum in Eq. (2).

$$E_M = \sum_{c \in C} E_c \cdot v_c \tag{2}$$

The network parameters $\boldsymbol{W}$ on the t-th iteration are updated using the gradient of the cost function with a training coefficient η, as shown in Eq. (3).

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta \frac{\partial E_M}{\partial \boldsymbol{W}_t} \tag{3}$$

From Eq. (2) and our definition that the cost function E_M is the

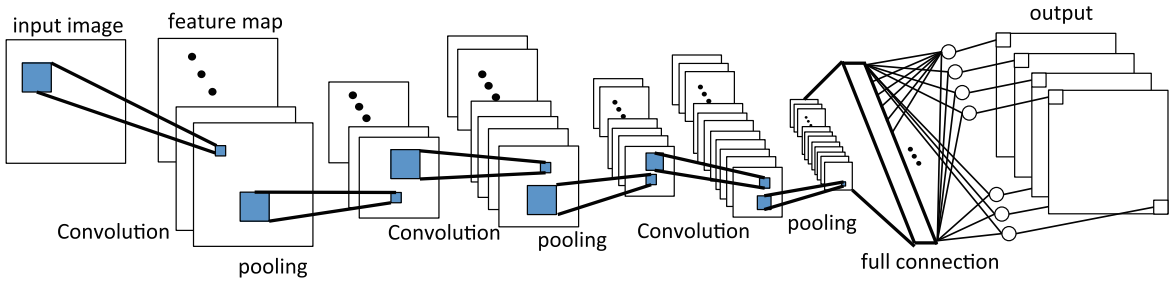

Fig. 2 Structure of DCNN for segmentation. The network consists of successive convolution and pooling layers followed by a full connection layer. The output layer has label maps for each class.

Table 1 The ratio of each face component in an image.

class	Ratio[%]
face	17.51
eyes	0.44
nose	1.08
mouth	0.89
eye glasses	0.06
hat	0.90
hair	10.73
others	68.34

sum of the errors across all classes, Eq. (3) can be written as Eq. (4).

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta\left(\frac{\partial}{\partial \boldsymbol{W}_t}E_0 \cdot v_0 + \frac{\partial}{\partial \boldsymbol{W}_t}E_1 \cdot v_1 + \cdots + \frac{\partial}{\partial \boldsymbol{W}_t}E_C \cdot v_C\right) \quad (4)$$

As shown in Eq. (4), the gradient of the cost function is the sum of the error gradients in all classes, and the errors in each class are independent. The product of the gradient of the cost function and the training coefficient η can be considered as the product of the error gradient for each class and the training coefficient. Therefore, Eq. (4) becomes:

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \left(\eta \cdot v_0 \cdot \frac{\partial E_0}{\partial \boldsymbol{W}_t} + \eta \cdot v_1 \cdot \frac{\partial E_1}{\partial \boldsymbol{W}_t} + \cdots + \eta \cdot v_C \cdot \frac{\partial E_C}{\partial \boldsymbol{W}_t}\right) \quad (5)$$

Both the cost weight of each class v_c and the training coefficient η are constants, but the cost weights are generally different for each class. As a result, the weighted cost function is equivalent to changing the coefficient for each class.

Table 1 gives the ratio of each class in the Labeled Face in the Wild (LFW) dataset. face and hair accounts for 17% and 10% of the whole image set, respectively, whereas the ratio of nose and eye regions are small (1.0% and 0.4%, respectively). The cost weight of each class is defined by Eq. (6) using the prior probability $p(c)$ computed from the ratios in Table 1.

$$v_c = \frac{C}{C - \log p(c)} \quad (6)$$

Note that C is the number of classes. The network parameters are updated according to the weighted cost function, and the convergence speed of a certain class is adjusted by its cost weight.

4. Experiments

We now demonstrate the performance of the proposed cost-alleviative learning with the weighted cost function. Although the LFW dataset has pixel-wise labeled data, there are only three classes, namely face, hair, and others. To prepare classes with low prior probabilities, we annotated new pixel-wise labels for eight classes, i.e., eyes, nose, mouth, sunglasses, hat, hair, face, and others. We split the dataset into 9,263 training images and 3,970 test images. We augmented the training subset by applying translation, rotation, and scaling to the images, giving a total of 185,262 images. **Table 2** lists the evaluation network structure and training parameters. The network has three convolution and pooling layers and one full connection layer with dropout.

Table 2 Network structure and training parameters.

layer		
input	gray	100×100
1st convolution	# of filters (size) activation function pooling (size)	32 (7×7) maxout maxpooling (2×2)
2nd convolution	# of filters (size) activation function pooling (size)	64 (6×6) maxout maxpooling (2×2)
3rd convolution	# of filters (size) activation function pooling (size)	128 (6×6) maxout maxpooling (2×2)
4th full connection	# of units activation function dropout	1,000 sigmoid 0.5
output	# of units	$8 \times 100 \times 100$
training parameters	batch size learning rate	10 0.1

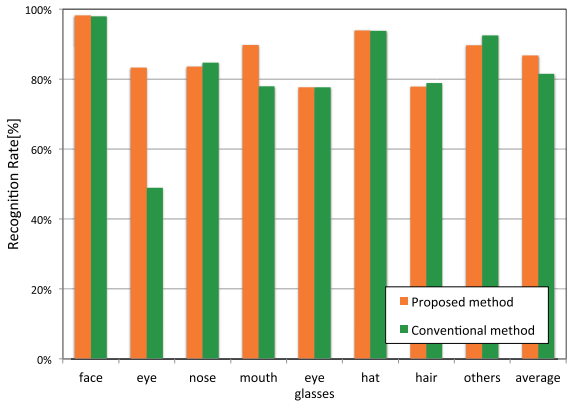

Fig. 3 Labeling accuracy of each facial part.

We employed maxout [4] and sigmoid activation functions in the convolution layer and full connection layer, respectively. The input images were 100×100 pixels in size, and the output units were $100 \times 100 \times 8$. We set the mini-batch size to 10 and applied a training coefficient of 0.1.

4.1 Face Part Labeling with Cost-alleviative Learning

We compared the recognition performance of our method with that of conventional DCNN. The cost function of our method includes weights for each class. The cost weights of the eyes, mouth, nose, face, and hair were found to be 0.78, 0.8, 1.0, 1.05, and 1.05, respectively. **Figure 3** shows the recognition performance after 1 million iterations. The recognition rates for face and hair are 98% and 78%, respectively. This is similar to the performance of conventional DCNN. Whereas the DCNN recognition rates for eyes and mouth are 49% and 78%, our method significantly improves these to 83% and 89% for each class. Our approach also improves the average recognition rate by about 5%.

Figure 4 illustrates the recognition performance during the training phase to demonstrate the behavior of the proposed cost function. The permanence of the face class does not influence the cost weight of the eye class, and is quite similar to that in the conventional method. The weight increases rapidly at first, and converges after about 100,000 iterations. In contrast, the eye class is not well recognized until the face region recognition has converged. The performance of the eye class increases gradually

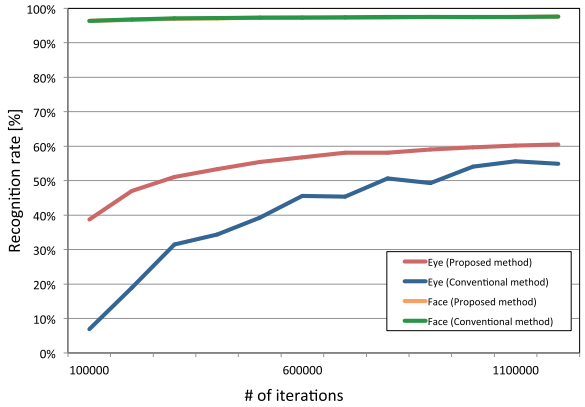

Fig. 4 Recognition rate of face and eye classes on each iteration.

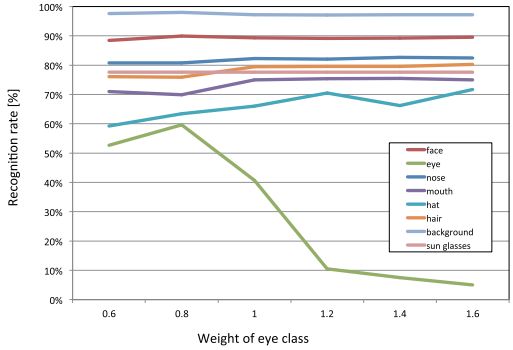

Fig. 5 Recognition rate of each class at each cost weight.

after 100,000 iterations, but remains at about 50%. The weighted cost function alleviates the error, and significantly improves the performance in earlier iterations.

4.2 Evaluation of Cost Weight Values

We define the cost weight from the prior probability of each class in Eq. (6). We now demonstrate the recognition behavior with manually set cost weights. We vary the cost weight of the eye class from 0.6 to 1.6, and train the network over 1 million iterations. **Figure 5** shows the recognition performance of each class. The recognition performance decreases when the cost weight of the eye class takes values larger than 1. However, the recognition performance is improved when the cost weight of the eye class is set at around 0.8, but gradually worsens with smaller cost weights. Thus, a smaller cost weight for the eye class (around 0.8) can improve the recognition performance. In Eq. (6), it is possible to set the cost weight of eyes to 0.78, which is almost the same as in this manual definition.

5. Discussion

Figure 6 shows the precision and recall curves of the eye and face classes. In addition, the results of face part labeling at each iteration are shown in **Fig. 7**. In the conventional method, the eye class exhibits high precision, but a low recall rate. This indicates that the class occupies a small region and is correctly labeled, but the labeled region is quite small. The eye class could not be labeled well because it is difficult to determine the boundary. While the recall rate gradually increases, many iterations are needed to obtain better performance. Our cost-alleviative learning achieves high precision and recall rates in fewer iterations by applying a

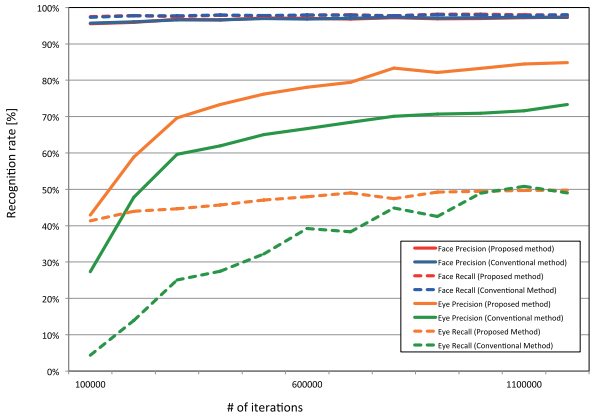

Fig. 6 Precision and recall curves of eye and face classes.

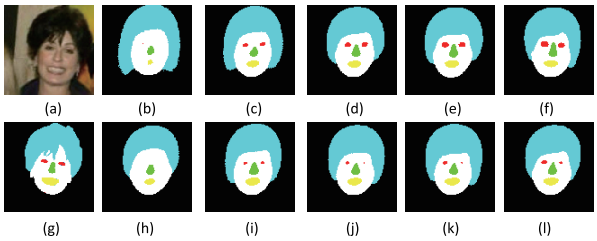

Fig. 7 Facial labeling results after different numbers of iterations. (a) is the input image, (g) is the ground truth, (b)–(f) are given by the proposed method and (h)–(l) are given by the conventional method. (b) and (h) are at 100,000 iterations, (c) and (i) are at 200,000 iterations, (d) and (j) are at 300,000 iterations, (e) and (k) are at 400,000 iterations, and (f) and (l) are at 500,000 iterations.

small cost weight for the eye class. As a result, it is easy to determine the boundaries, leading to a high precision rate. Moreover, the recall rate converges after fewer iterations. The face region retains high recall and precision rates, and does not influence the cost weight of the eye class. Thus, cost-alleviative learning enables an efficient improvement in recognition performance and convergence rate.

6. Conclusion

We have proposed a cost-alleviative learning method that uses a weighted cost function to improve the recognition performance of certain classes during semantic segmentation. The weighted cost function enables the training coefficient for each class to be adjusted. This alleviates the error in classes that occupy small regions, making it possible to achieve convergence for such classes. In addition, the boundaries of each class may be recognized after fewer iterations, which will improve the performance. In facial part labeling, the recognition performance of the eye class can be significantly improved using cost-alleviative learning. In future work, we will apply the weighted cost function to scene parsing or other semantic segmentation tasks.

References

[1] Cootes, T., Taylor, C. and Graham, J.: Active shape models their training and application, *Computer Vision and Image Understanding*, Vol.61, No.1, pp.38–59 (1995).
[2] Cristinacce, D. and Cootes, T.: Boosted regression active shape models, *BMVC*, pp.1–10 (2007).
[3] Farabet, C., Couprie, C., Najman, L. and LeCun, Y.: Learning Hierarchical Features for Scene Labeling, *PAMI*, Vol.35, No.8, pp.1915–1929 (2013).

[4] Goodfellow, I.J., Farley, D.W., Mirza, M., Courville, A. and Bengio, Y.: Maxout Networks, *ICML*, pp.1319–1327 (2013).
[5] Krizhevsky, A., Sutskever, I. and Hinton, G.E.: Imagenet classification with deep convolutional neural networks, *NIPS*, pp.1097–1105 (2012).
[6] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P.: Gradient-based learning applied to document recognition, *IEEE*, Vol.86, No.11, pp.2278–2324 (2003).
[7] Liang, L., Wen, F., Xu, Y., Tang, X. and Shum, H.: Accurate face alignment using shape constrained Markov network, *CVPR*, pp.1313–1319 (2006).
[8] Liang, L., Xiao, R., Wen, F. and Sun, J.: Face alignment via component-based discriminative search, *ECCV*, pp.72–85 (2008).
[9] Luo, P., Wang, X. and Tang, X.: Hierarchical Face Parsing via Deep Learning, *CVPR*, pp.2480–2487 (2013).
[10] Milborrow, S. and Nicolls, F.: Locating facial features with an extended active shape model, *ECCV*, pp.504–513 (2008).
[11] Smith, B.M., Zhang, L., Brandt, J., Lin, Z. and Yang, J.: Exemplar-Based Face Parsing, *CVPR*, pp.3484–3491 (2013).
[12] Tu, Z., Chen, X., Yuille, A. and Zhu, S.: Image parsing: Unifying segmentation, detection and recognition, *IJCV*, Vol.63, No.2, pp.113–140 (2005).
[13] Zhou, Y., Zhang, W., Tang, X. and Shum, H.: A Bayesian mixture model for multi-view face alignment, *CVPR*, Vol.2, pp.741–746 (2005).
[14] Valstar, M., Martinez, B., Binefa, X. and Pantic, M.: Facial point detection using boosted regression and graph models, *CVPR*, pp.2729–2736 (2010).
[15] Warrell, J. and Prince, S.: Labelfaces: Parsing facial features by multiclass labeling with an epitome prior, *ICIP*, pp.2481–2484 (2009).

(Communicated by *Hitoshi Imaoka*)