CLUSTER CONVOLUTIONAL NEURAL NETWORKS FOR FACIAL AGE ESTIMATION

Chong Shang, Haizhou Ai

Tsinghua National Lab for Info. Sci. & Tech., Depart. of Computer Sci. & Tech.,
Tsinghua University, Beijing, China.
shang-c13@mails.tsinghua.edu.cn, ahz@mail.tsinghua.edu.cn

ABSTRACT

In computer vision tasks, such as age estimation problem, a typical deep learning model is insufficient to represent all the transformations between data and their labels. In this paper, we present a novel deep neural network called cluster convolutional neural network (Cluster-CNN) to estimate ages from facial images, which is based on the concept that clustering rich CNN features is able to assist the network to efficiently tackle the nonlinearity of this task. In particular, given a facial image, we first coarsely normalize a face to a standard size according to the distance between two eyes. Then the normalized face is fed into a Cluster-CNN to get the prediction. The cluster module we propose in this paper is able to capture multimodal transformations, and is differentiable, thus making it capable to be optimized in a unified backpropagation fashion. We evaluate the proposed method on the popular MORPH II dataset. To the best of our knowledge, our model outperforms all the other previous methods by a large margin.

***Index Terms*—** Age estimation, cluster CNN, face age

1. INTRODUCTION

Facial age estimation, *i.e.*, estimating the age from a facial image, plays an important role in computer vision task because of the booming of social networks and applications in security. It still remains a very challenging problem due to many difficulties, such as illumination variation, makeups and different face poses and diverse backgrounds, which enormously increase the nonlinearity and multi-modality in the task.

To solve these problems, finding distinguish features that can represent correlations between face images and their corresponding ages is essential. Some existing methods used hand-crafted features, such as Gabor in [1] and biological inspired features in [2], and adopted a classifier or a regressor to get final predictions. Unfortunately, the performances of these methods were limited by the representation ability of hand-crafted features. Inspired by remarkable performance improvements in the object classification, detection and the face recognition aiding by deep learning models, facial age estimation also benefited from deeply learned hierarchical features [3–6]. CNNs are able to exploit numerous training data, and learn a complex nonlinear transformation that tackle variations of illumination, pose and background well. These methods have achieved an excellent performance in this task.

However, we argue that the complicated relationships in the data have not been fully explored. Although CNNs are powerful models that are designed to handle transition invariant, we find out further improvements still can be achieved by dividing the CNN feature space into subspaces and learning different representations respectively.

In this paper, we consider the multi-modality in the age estimation task, and propose a novel module to equip CNNs with explicit cluster power. The idea behind Cluster-CNN is *divide-and-conquer*. Intuitively, a single branch of neurons that treats all samples equally is insufficient to tackle all the diversified variations, thus we adopt multiple branches that enable samples to choose a more suitable path to get the final prediction. For instance, an image taken in a dark room and one under bright sunshine should be treated separately. In addition, the powerful Cluster-CNN liberates the system from elaborate preprocessing, such as face alignment, making the model more robust. A similar idea is used in [7], where they attempted to cluster CNN features for metric learning. The significant difference is that instead of just grouping features we also maximize the margins between these clusters, leading to a more discriminative feature space, which is described in section 3.2.

Our contributions are in three-fold:

- We propose a novel Cluster-CNN to tackle the complex transformations in age estimation task, which can be trained in the backpropagation fashion.
- We evaluate different factors, such as normalization, solvers, joint learning and the proposed cluster module, and their importance to the final age prediction.
- We show that without elaborate preprocessing we can still get a relative good result.

2. RELATED WORK

Early approaches [1, 8–12] for predicting facial ages rely on hand-crafted features which are followed by a classifier or a

978-1-5090-2175-8/17/$31.00 ©2017 IEEE

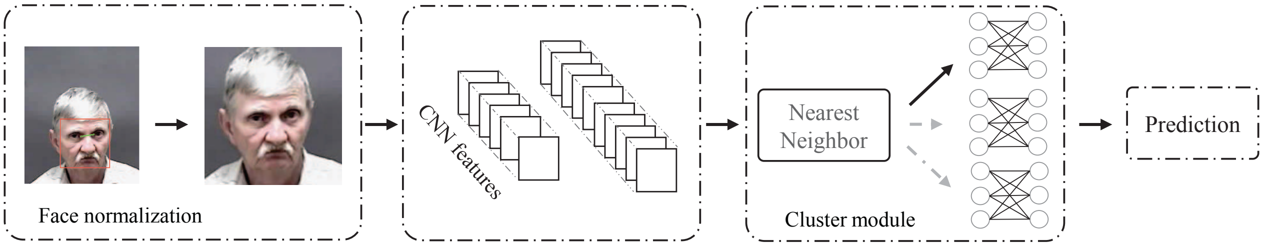

Fig. 1: **The proposed framework.** For a facial image, our method normalizes it with a landmark detector, then computes CNN features, chooses a branch with a learnable cluster module and finally gets a prediction. The solid arrow in the cluster module indicates the chosen branch.

regressor. Geng *et al.* [12] explored label distribution which allowed minor mistakes between similar ages. Chao *et al.* [8] made use of support vector regression (SVR) to regress ages in a local space. A more elaborately designed feature can be found in [2], which adopted biological inspired features to mine the age-related information in a face image. These feature are of critical importance and are carefully designed for capture age-related appearance in a face image.

As boosting improvements in object classification and detection continue emerging by using deep learning models, some works [3–6, 13] adopted CNNs for age estimation. Levi *et al.* [5] directly applied a common CNN to classify face images into different age ranges instead of predicting an accurate age. Liu *et al.* [14] further studied the roles of different facial parts (*e.g.*, mouth, eyes and nose) in this task. Wang *et al.* [4] treated the CNN as a feature extractor which was followed by a SVM and a SVR. A more recent work Rothe *et al.* [3] adopted a deeper network to vote for the final prediction, and made a large progress in performance. Our approach, shown in Fig. 1, no need of elaborate preprocessing, outperforms these methods mentioned above by a large margin.

3. METHODOLOGY

3.1. Face normalization

To compensate for pose and scale variations, the input face image is normalized to a standard scale according to the distance between two eyes, which is computed by a typical landmark detector. There are many face landmark detection methods [15] that are well enough for this preprocessing. We crop and normalize the face image using a fixed ratio α to the eye distance l. In the training stage, we add a small noise denoted as ϵ in training samples to make the model robust to scale variation. Thus we can get the face area $w = (\alpha + \epsilon) \times l$. We then crop the face area and resize it to a fixed size.

3.2. Cluster CNN

3.2.1. Model definition

In each layer of a typical CNN there are multiple convolution kernels called network width, which are the key module to extract rich features of many different aspects, *e.g.*, some kernels respond to edges and some to textures. These CNN features are then fed into a fully connected network to get the final outputs. To further improve the capacity of this model, we propose to divide the features into groups and learn different representations separately.

Such a cluster module should be differentiable so that it can be optimized as we train the CNN. We consider to use k-means++ to initially cluster these examples and get the centroids of each cluster. Then we can assign each sample into the nearest cluster. If we record the assignments of these examples, we are able to apply backpropagation through the network. Thus we can get a differentiable cluster module in a traditional CNN.

We use x to denote features, v the output of this module, c_i the nearest cluster, u_{c_i} the centroid of c_i, W_i, b_i the corresponding distinct weights, ϕ the activation function. Then the cluster module can be defined as,

$$v = \phi(W_i(x - u_{c_i}) + b_i), \;\; x \in c_i.$$

3.2.2. Training with Monte Carlo sampling

Note that the model is differentiable to x. In the training stage we minimize the distance within a cluster, denoted as S_w,

$$S_w = \frac{1}{N}\sum_i \|x_i - u_{c_{x_i}}\|, \;\; x_i \in c_{x_i}.$$

To maximum the margins between two clusters, we add a max margin term to the objective, denoted as S_b,

$$S_b = \frac{1}{N_c}\sum_i \|u_{c_i} - u_{c_j}\|, \;\; i \neq j.$$

Following the idea of LDA algorithm, we rewrite the final objective as

$$min \, \frac{S_w}{S_b}. \tag{1}$$

We describe how to *hard*-assign samples to each cluster above. However, soft assign is often more robust, since it draws a distribution for the original features and allows 'noise' (outliers) in the process. We compute the probability that an example falls into a certain cluster, and sample a cluster according to the probability distribution. The probability is defined as

$$p(c_i \mid x, C) = \frac{\|x - u_c\|^{-2}}{Z}, \tag{2}$$

where $Z = \sum_c (x - c)^{-2}$ normalizes the probability.

The training strategy we use to optimize the cluster model is described in Algorithm 1. As batch normalization described in [16], we also update the centroids during the training stage, a new centroid is computed as,

$$u^*_{c_i} = \alpha u_{c_i} + (1 - \alpha)x, \;\; x_i \in c_i, \tag{3}$$

where α is a momentum that controls the updating rate and is set to 0.999 in the experiments.

Algorithm 1 Training Cluster-CNN

```
Input: Number of clusters $k$, CNN features $x$ in a mini-batch
Output: Cluster sets denoted as $C$
procedure CLUSTER-CNN –TRAINING
    Initialize: $C$ using k-means++
    for each example $x$ in a mini-batch do
        compute $p(c \mid x, C)$ using Equation 2
        draw $c_i \sim p(c \mid x, C)$
        assign x to $c_i$
        update $c^*_i = \alpha c_i + (1 - \alpha)x$
        forwardpropagation $x$ through the $c_i$ branch
        compute the objective using Equation 1
        backpropagation through $c_i$ branch
    end for
end procedure
```

3.3. Network optimization

We adopt GoogLeNet [17] pre-trained on Imagenet [18] to extract CNN features, since it contains many different sizes of kernels which is able to extract features from multiple scale levels. Then the features are fed into the cluster module described above. In particular, since the mini-batch is relatively small (*e.g.*, 128), we cache CNN features as many as needed (*e.g.*, 5,000). Then we perform k-mean++ algorithm to cluster these features into k groups. For each group, we train a distinct branch which represents a unique transformation.

For the final objective, we minimize both the cross-entropy error $E_c = -\log p(y_i = l_i \mid x_i)$ and the regression error $E_r = \frac{1}{N}\sum_i \|l_i - y_i\|$. The final loss is computed as

$$E = \lambda E_c + (1 - \lambda)E_r,$$

where λ balances the loss weights, and x_i, l_i, y_i is the i-th example, the label and the prediction respectively.

Table 1: Statistics of MORPH II dataset.

	< 20	$20+$	$30+$	$40+$	$50+$	Total
Male	$6,638$	$14,016$	$12,447$	$10,062$	$3,482$	$46,645$
Female	831	$2,309$	$2,910$	$1,988$	451	$8,489$
Total	$7,469$	$16,325$	$15,357$	$12,050$	$3,933$	$55,134$

During the training time, the model is optimized with Adam solver [19] in an end-to-end fashion, and the initial learning rate is set to 0.0001. The input images are resized into 128×128 to feed to the network. And the network is trained for nearly 5 epochs.

4. EXPERIMENTS

To show the efficiency of the proposed method, we evaluate our method on the popular age estimation dataset FG-NET [20] and MORPH II [21]. FG-NET consists 1,002 images of 82 persons. The ages range from 0 to 69 years old. MORPH II contains 55,000 images of over 13,000 persons. And the ages range widely from 16 to 77 with a average of 33. This dataset provides annotations including race, gender and age. Here, we focus on predicting ages, and ignore other labels. Table 1 shows a statistic of this dataset.

To compare with existing methods, we adopt the same experimental setting as the works in [3]. MORPH II is randomly divided into 80% for training and 20% for testing. And we repeat it for five times. For FG-NET, we also adopt 'leave-one-person-out' cross validation like [3, 4, 12]. We also adopt the mean absolute error (MAE) as evaluation metrics.

The experimental results is shown in Table 2. We evaluate the effects of different modules to the performance, including pre-trained parameters, choice of solvers, classification, face normalization and cluster modules. There is a significant improvement if the network is pre-trained on other tasks, such as classification on Imagenet, which may be due to better initialization of the network. In this subsection, we will show the importance of each module.

4.1. Results

4.1.1. A strong baseline

We start from a baseline method. We adopt GoogLeNet architecture as described in [17] with the pre-trained parameters on Imagenet. After resizing them into 128×128, we feed the original image without normalization into the network to train a baseline model. This network already outperforms the state-of-the-art methods, to the best of our knowledge. The performance improvement might be due to the powerful CNN architecture such as the inception module, and the Adam solver.

Table 2: Comparison of different factors on MORPH.

						k=5	k=3	k=5	k=7
Pre-train?		✓	✓	✓	✓	✓	✓	✓	✓
Adam?			✓	✓	✓	✓	✓	✓	✓
Cls?				✓	✓	✓	✓	✓	✓
Normed?					✓		✓	✓	✓
Cluster?						✓	✓	✓	✓
MAE	6.72	3.20	3.18	3.16	2.82	2.86	2.75	**2.71**	**2.70**

4.1.2. Face normalization

Then we add the face normalization described in 3.1 to compensate for pose and scale variations. To avoid landmark failures, we add a random noise to the cropping scale in the training samples. With this preprocessing, the performance is improved by a large margin of 0.34 MAE.

4.1.3. Cluster module

As shown in Table 2, the proposed Cluster CNN can fill the performance gap of normalization. We set k to 5, use $5,000$ examples to initialize the cluster centroids and train the Cluster-CNN with the same settings. By adding this module to the baseline model alone, we achieve the similar result as face normalization, which shows the cluster module can efficiently handle the variations and is a substitute for preprocessing.

Then we combine all the bells and whistles in the proposed method and lead to a more accurate result. To trade off between accuracy and complexity of the model we set $k = 5$. To the best of our knowledge, which is a significant improvement in this task.

4.2. Comparison with other methods

In Table 3, we compare our approaches with existing methods.

MORPH II. Works [2, 8, 12, 22, 23] that are based on handcrafted features are listed above. Compared to deep learning models, these approaches suffers from model limitations. Works [4, 14, 24, 25] that adopted relatively shallower CNNs might lack of representation abilities. Rothe *et al.* [3] exploited VGG architecture [26], and used classification to predict ages, which adopted the similar experimental setting as our baseline model. However, our baseline model achieves a slightly better result, which is due to the multi-scale convolution kernels in GoogLeNet. Moreover, equipped with the proposed cluster module our method achieves a more accurate result on both datasets.

FG-NET. Images from FG-NET are well aligned, thus we do not apply face normalization on this dataset. Since this dataset contains only 1,002 images, the model easily tends to overfit on the training set. To alleviate this effect, we reduce the model size. In particular, we downsample input images to 72×72 and reduce the depth of the CNN by half. We achieve only a slightly better result by adding the cluster module. This can be explained by the fact due to the lack of data a deep learning model overfits to noises on the training set, and gives poor results on the test set. We also show that more accurate predictions can be achieved by pre-training on the MORPH dataset.

Table 3: Comparative results with other methods.

Method	MORPH	FG-NET
Human performance [2]	6.30	4.70
Chang *et al.* [22]	6.07	4.48
Chen *et al.* [23]	5.88	4.67
Guo *et al.* [11]	N/A	5.07
Geng *et al.* [12]	4.87	4.76
Chao *et al.* [8]	N/A	4.38
Han *et al.* [2]	3.80	4.80
Wang *et al.* [4]	4.77	4.26
Feng *et al.* [6]	4.59	4.35
Guo *et al.* [25]	3.92	N/A
Yi *et al.* [24]	3.63	N/A
Liu *et al.* [14]	3.48	N/A
Rothe *et al.* [13]	3.45	5.01
Rothe *et al.* [3]	3.25	4.63
Ours - baseline	3.16	4.35
Ours - cluster	2.86	**4.15**
Ours - cluster (more data)	N/A	**3.85**
Ours - proposed	**2.71**	N/A

5. CONCLUSION

In this paper, we aim to tackle difficulties in age estimation task. We propose a differentiable Cluster-CNN module to cluster CNN features into different subspaces, which is able to further handle multimodal patterns. In the training stage, we also maximize the margins between clusters, leading to a more discriminative feature space. Moreover, we evaluate different factors in this task, such as solvers, the face normalization, and their corresponding effects to the final prediction. We show that a strong baseline can be obtained without elaborate preprocessing. With a rough face normalization and a cluster module our approach outperforms existing methods by a large margin. However, more mechanisms can be adopted to further improve age estimation performance, such as a fine face alignment. We will leave this to the future work.

6. ACKNOWLEDGMENT

This work was supported by the Natural Science Foundation of China (Project Number 61521002).

7. REFERENCES

[1] Feng Gao and Haizhou Ai, "Face age classification on consumer images with gabor feature and fuzzy lda method," in *ICB*, 2009.

[2] Hu Han, Charles Otto, Xiaoming Liu, and Anil K Jain, "Demographic estimation from face images: Human vs. machine performance," *TPAMI*, 2015.

[3] Rasmus Rothe, Radu Timofte, and Luc Van Gool, "Deep expectation of real and apparent age from a single image without facial landmarks," *IJCV*, 2016.

[4] Xiaolong Wang, Rui Guo, and Chandra Kambhamettu, "Deeply-learned feature for age estimation," in *WACV*, 2015.

[5] Gil Levi and Tal Hassner, "Age and gender classification using convolutional neural networks," in *CVPR Workshops*, 2015.

[6] Songhe Feng, Congyan Lang, Jiashi Feng, Tao Wang, and Jiebo Luo, "Human facial age estimation by cost-sensitive label ranking and trace norm regularization," *Transactions on Multimedia*, 2017.

[7] Oren Rippel, Manohar Paluri, Piotr Dollar, and Lubomir Bourdev, "Metric learning with adaptive density discrimination," in *ICLR*, 2016.

[8] Wei-Lun Chao, Jun-Zuo Liu, and Jian-Jiun Ding, "Facial age estimation based on label-sensitive learning and age-oriented regression," *Pattern Recognition*, 2013.

[9] Eran Eidinger, Roee Enbar, and Tal Hassner, "Age and gender estimation of unfiltered faces," *TIFS*, 2014.

[10] Xin Geng, Zhi-Hua Zhou, and Kate Smith-Miles, "Automatic age estimation based on facial aging patterns," *TPAMI*, 2007.

[11] Guodong Guo, Yun Fu, Charles R Dyer, and Thomas S Huang, "Image-based human age estimation by manifold learning and locally adjusted robust regression," *TIP*, 2008.

[12] Xin Geng, Chao Yin, and Zhi-Hua Zhou, "Facial age estimation by learning from label distributions," *TPAMI*, 2013.

[13] Rasmus Rothe, Radu Timofte, and Luc Van Gool, "Some like it hot-visual guidance for preference prediction," in *CVPR*, 2016.

[14] Ting Liu, Jun Wan, Tingzhao Yu, Zhen Lei, and Stan Z Li, "Age estimation based on multi-region convolutional neural network," in *CCBR*, 2016.

[15] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun, "Face alignment by explicit shape regression," *IJCV*, 2014.

[16] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[19] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] Gabriel Panis, Andreas Lanitis, Nicholas Tsapatsoulis, and Timothy F Cootes, "Overview of research on facial ageing using the fg-net ageing database," *IET Biometrics*, 2016.

[21] Karl Ricanek and Tamirat Tesafaye, "Morph: A longitudinal image database of normal adult age-progression," in *FGR*, 2006.

[22] Kuang-Yu Chang, Chu-Song Chen, and Yi-Ping Hung, "Ordinal hyperplanes ranker with cost sensitivities for age estimation," in *CVPR*, 2011.

[23] Ke Chen, Shaogang Gong, Tao Xiang, and Chen Change Loy, "Cumulative attribute space for age and crowd density estimation," in *CVPR*, 2013.

[24] Dong Yi, Zhen Lei, and Stan Z Li, "Age estimation by multi-scale convolutional network," in *ACCV*, 2014.

[25] Guodong Guo and Guowang Mu, "A framework for joint estimation of age, gender and ethnicity on a large database," *IVC*, 2014.

[26] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *CVPR*, 2015.