Signal Processing 144 (2018) 384–391

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

A Deep Residual convolutional neural network for facial keypoint detection with missing labels☆

Shaoen Wu[a,b,1,*], Junhong Xu[b], Shangyue Zhu[b], Hanqing Guo[b]

[a] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China
[b] Department of Computer Science, Ball State University, Muncie, IN 47306, USA

ARTICLE INFO

Article history:
Received 19 April 2017
Revised 29 August 2017
Accepted 2 November 2017
Available online 4 November 2017

Keywords:
Deep learning
Facial keypoint detection

ABSTRACT

Keypoint detection is critical in image recognitions. Deep learning such as convolutional neural network (CNN) has recently demonstrated its tremendous success in detecting image keypoints over conventional image processing methodologies. The deep learning solutions, however, heavily rely on labeling target images for their reliability and accuracy. Unfortunately, most image datasets do not have all labels marked. To address this problem, this paper presents an effective and novel deep learning solution, *Masked Loss Residual Convolutional Neural Network* (ML-ResNet), to facial keypoint detection on the datasets that have missing target labels. The core of ML-ResNet is a *masked loss objective function* that ignores the error in predicting the *missing* target keypoints in the output layer of a CNN. To compensate for the loss induced by the masked loss objective function that likely results in overfitting, ML-ResNet is designed of a data augmentation strategy to increase the number of training data. The performance of ML-ResNet has been evaluated on the image dataset from Kaggle Facial Keypoints Detection competition, which consists of 7049 training images, but with only 2140 images that have full target keypoints labeled. In the experiments, ML-ResNet is compared to a pioneer literature CNN facial keypoint detection work. The experiment results clearly show that the proposed ML-ResNet is robust and advantageous in training CNNs on datasets with missing target values. ML-ResNet can improve the learning time by 30% during the training and the detection accuracy by eight times in facial keypoint detection.

1. Introduction

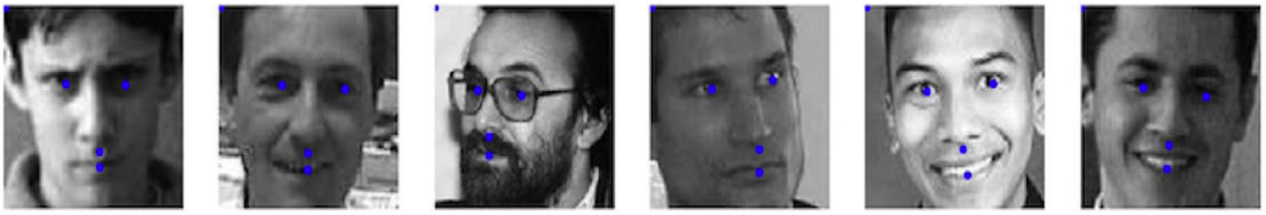
Detection of facial keypoints is challenging because of variations of faces, light exposures, different viewpoints, etc. In face keypoint detection, it is essential to analyze facial expressions and track faces. Recent development of CNNs has shown great success in computer vision [1–5]. The deep structures of CNNs extract raw data into a high level abstraction, which consists of various aspects of an input that keep the distinguished features but discard irrelevant information [5]. Traditional image processing based works on facial keypoint detection are commonly based on searching local image features [6,7]. Specifically, each keypoint is detected by a classifier called *component detector* based on local patches. As a result, local minima may be incurred by ambiguous or corrupted local patches. Recent approaches address the problem of local minima by employing cascading CNNs to facial keypoints detection. Sun, Wang and Tang propose to use several CNNs at different levels to predict and finely tune facial keypoint positions [8]. Another CNN based solution predicts keypoints with data augmentation [9] and it has shown significant improvements in Labeled Faces in the Wild(LFW) dataset [10]. Most CNN solutions in literature, however, have *a common issue in using CNNs as the predictor that they are not able to be trained on samples that have missing target values*. Many facial images unfortunately do not have all facial keypoints available because some angles of viewing faces can result in undisclosed keypoints. Excluding these images will significantly reduce the number of sample images available for the training of deep neural networks that requires a large amount of data to prevent overfitting. Therefore arises a research challenge that how to fit into deep neural networks the facial images of missing keypoints if they are retained in the training dataset to prevent the overfitting.

In this work, we design and evaluate an effective solution, named Masked Loss Residual ML-ResNetConvolutional Neural Network (ML-ResNet), that proposes a novel objective function to address the above research challenge. ML-ResNet adds a mask matrix at the output layer of the CNN to mask out the predicted value

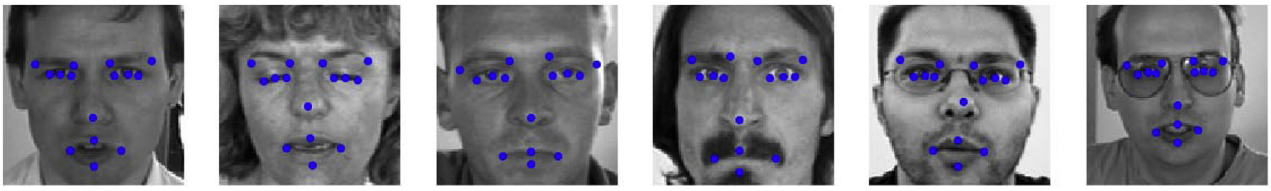
☆ This paper is an extended version of the work accepted by Mobimedia 2017.

* Corresponding author.
E-mail address: swu@bsu.edu (S. Wu).
[1] Senior Member, IEEE

https://doi.org/10.1016/j.sigpro.2017.11.003
0165-1684/



(a) Missing keypoints images.



(a) Fully labeled images.

Fig. 1. In the Kaggle dataset, some of the faces have keypoints fully labeled, but some not.

at the same index that expects to be a missing target value. As a result, the error between the target value and predicted value will not be affected by the missing value. The neural network will not be updated by the back-propagation on missing target values. In addition to the proposed objective function, we also design a deep residual convolutional neural network (ResNet) to replace the conventional CNN. Introducing an identity mapping shortcut connection [3], the ResNet can prevent the problem degradation and allow to be trained on deeper networks. We furthermore adapt a recent technique, *batch normalization* [11], to avoid internal covariate shift. We evaluated the performance of our solution ML-ResNet and a traditional CNN solution [9] with the Facial Keypoints Detection dataset hosted at Kaggle that contains 7094 training images, but with only 2140 images that have all target keypoints labeled. In the Kaggle dataset, each image sample is a grayscaled image with 96×96 pixels with 15 facial keypoints to predict. Some samples are shown in Fig. 1. In the training phase, a data augmentation method [9] is employed to improve the generalization of the two models. The results show that the performance of our ML-ResNet largely surpasses the traditional CNN when trained on missing target keypoints dataset. In the rest of this paper, Section 2 introduces the model of an image in a CNN network and the related solutions in literature that employ deep neural networks to address facial keypoints detection. Then, Section 3 discusses the detail design of the proposed solution. The performance evaluation is next presented in Section 4. Finally the paper is concluded by Section 5.

2. Background

In this section, we first introduce how an image is modeled and computed by a CNN, followed by the a summary of related work proposed for facial keypoint detection.

2.1. Image model in CNN

Images is often stored in a 3-dimensional array, where dimensions (h, w, c) represent the height, width, and color channel of an image. Because the spatial positions of pixels are important, convolutional layers in CNNs retain this spatial position information by performing a 2-dimensional convolution operation along h and w axes. The input and output of each convolutional layer constitute a *feature map*, which is a 3-dimensional array with the size of $h \times w \times c$, where h and w are spatial dimensions representing

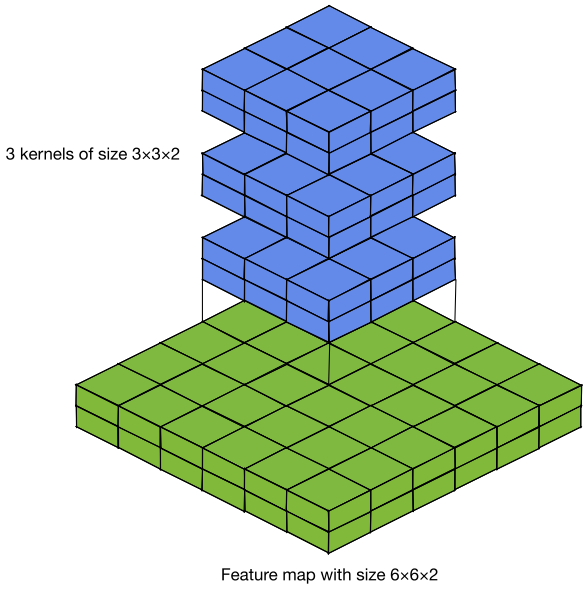


Fig. 2. Demonstration of kernels and feature map in a convolutional layer. Three kernels with the size of $3 \times 3 \times 2$ computing on a feature map of the size of $6 \times 6 \times 2$.

height and width and c is the color depth. The input to the first convolutional layer is an image of $h \times w$ pixels with c color channels (for gray-scale image, c is 1 and RGB image, c is 3). Each convolutional layer consists of n learnable *kernels* of size $h' \times w' \times c$ representing height, width, and depth. The height h' and width w' are usually small to learn local features. The depth is the same as the input feature map. An illustration is plotted in Fig. 2. On the figure, the size of the input *feature map* is $6 \times 6 \times 2$. Three *kernels* with a dimension of $3 \times 3 \times 2$ convolve through the *feature map*. The size of the output *feature map* is decided by two parameters *stride s* and *padding p*. *Stride s* indicates the distance between two consecutive positions in the matrix multiplication between each *kernel* and local features in the *feature map*. *Padding p* represents how many 0 valued pixels to pad along h and w axes of the input *feature map* to maintain the size of the output *feature map*. The output size is calculated as in Eq. (1), where l indicates the layer number, h, w, c represent the height, width, and depth of the fea-