Object Specific Deep Feature for Face Detection

SUMMARY Motivated by the observation that certain convolutional channels of a Convolutional Neural Network (CNN) exhibit object specific responses, we seek to discover and exploit the convolutional channels of a CNN in which neurons are activated by the presence of specific objects in the input image. A method for explicitly fine-tuning a pre-trained CNN to induce object specific channel (OSC) and systematically identifying it for the human faces has been developed. In this paper, we introduce a multi-scale approach to constructing robust face heatmaps based on OSC features for rapidly filtering out non-face regions thus significantly improving search efficiency for face detection. We show that multi-scale OSC can be used to develop simple and compact face detectors in unconstrained settings with state of the art performance.

key words: deep feature, convolutional neural network, object specific channel, face detection

1. Introduction

Convolutional Neural Networks (CNNs) have been demonstrated to have state-of-the-art performances in many computer vision tasks such as image classification [1], detection [2], retrieval [3] and captioning [4]. Several methods such as DeepVis [5] and deconvolutional technique [6] have been developed to help get a better understanding of how these models work and it has been shown that CNNs have the ability to learn powerful and interpretable features.

In this paper, we present a method for discovering and exploiting object specific deep learning features and use face detection as a case study. A key motivation of this paper is based on the observation that certain convolutional channels of CNNs exhibit object specific responses [5]. An object specific channel (OSC) is a convolutional feature map at a hidden layer of a CNN, in which neurons are strongly activated by the presence of a certain class of objects at the neurons’ corresponding regions in the input image. An example is shown in Fig. 1 where the last image at the top row is a face specific OSC, in which spatial locations corresponding to the face regions have strong responses (white pixels) while areas corresponding to non-face regions have weak responses (black pixels). If such an OSC can be reliably identified, then it can be exploited for various tasks including object detection. Do such channels exist for a given class of objects? If so, how can we systematically identify such channels? If not, can we tune a pre-trained CNN to have such channels?

A method for explicitly fine-tuning a pre-trained CNN to induce an OSC and systematically identifying it for the human face object has been developed. Based on the basic OSC features, we introduce a multi-scale approach to constructing robust face heatmaps for fast face detection in unconstrained settings. Our main contribution is to propose the object specific channel (OSC) based on pretrained deep convolutional neural network and systematically identify it for the human faces. We show that OSC features can be used for rapidly filtering out non-face regions to build compact face detector with state of the art performance.

2. Related Work

Face detection models in the literature can be divided into four categories: Cascade-based model, Deformable Part Models (DPM)-based model, Exemplar-based model and Neural-Network-based model. The most famous cascade-base model is the VJ detector [7] based on Haar-like features, which have demonstrated excellent performance for frontal face detection. However Harr-like features have limited representation ability to deal with variational settings. Some works try to improve VJ-detector via using more complicated features such as SURF [8], HoG [9] and polygonal Haar-like features [10]. Aggregate channel features [11] are also introduced for solving multi-view face detection problems.

Another category is DPM-based model [12], which treats face as a collection of small parts. DPM-based model can benefit from the fact that different facial parts independently have lower visual variations, so it is reasonable to build robust detectors by combining different models trained for individual parts. For example, Part-based structural models [13] have achieved success in face detection and a vanilla DPM can achieve top performance over the more sophisticated DPM variants [14].

Exemplar-based detectors [15] try to bring image retrieval techniques into face detection to avoid explicitly modeling different face variations in unconstrained settings. Specifically, each exemplar casts a vote following the Bag-of-Words (BOW) [16] retrieval framework to get a voting map and uses generalized Hough Voting [17] to locate the
faces in the input image. As a result, faces can be effectively detected in many challenging settings. However, a considerable amount of exemplars is required to cover all kinds of variations.

Neural-Networks-based detectors are usually based on deep convolutional neural networks. Faceness [18] tries to find faces through scoring facial parts’ responses by their spatial structure and arrangement, and different facial parts correspond to different CNNs. A two-stage approach is also proposed by combining multi-patch deep CNNs and deep metric learning [19]. The CCF detector [20] uses an integrated method called Convolutional Channel Features, transferring low-level features extracted from pre-trained CNN models to a boosting forest model. Cascade architectures based on CNNs [21] have been also designed to help reject background regions at low resolution, and select face area carefully at high resolution. The DDFD detector [22] uses a single model based on deep convolutional neural networks for multi-view face detection, and points out that CNNs can benefit from better sampling and more sophisticated data augmentation techniques.

We try to directly use a single convolutional channel to produce face response heatmap, which can be used to quickly locate potential face area. The heatmap is similar to the voting map in exemplar-based approach [15], but the difference is that their voting map is produced by the Bag-of-Words (BOW) [16] retrieval framework, while our heatmap is directly extracted from a convolutional channel.

3. **Our Method**

3.1 Method Overview

Our goal is to discover and exploit face specific convolutional channel (features) to help locate the face areas quickly for further processing. Our system contains two stages as shown in Fig. 1: In the first stage, the face heatmap of a face image can be generated by face specific channel in a trained CNN with a multi-scale approach. Thresholding the heatmap quickly filters out non-face regions, dramatically reduces face search space without throwing away genuine face regions. In the second stage, a set of face candidate windows can be quickly identified based on the heatmap, and all candidates are then processed by a CNN based binary classifier. Finally all face windows are merged using Non-Maximum Suppression (NMS) [23] to obtain the final detection results.

3.2 Face Specific CNN Channel Extraction

3.2.1 Training Data Preparation and CNN Fine-Tuning

We start with the pre-trained “AlexNet” [1] provided by the open source Caffe Library [25]. In order to adapt the CNN model to our face detection problem, we change the last classification layer from ImageNet-specific 1000 classes to 2 classes, which represent images with faces and images with masked faces (Fig. 2) respectively. Specifically, let Conv be a convolutional layer, LRN a local response normalization layer, P a max pooling layer and F a fully connected layer, the architecture can be described as Conv1(55 × 55 × 96) - LRN - P - Conv2(27 × 27 × 256) - LRN - P - Conv3(13 × 13 × 384) - Conv4(13 × 13 × 384) - Conv5(13 × 13 × 256) - P - F(4096) - F(4096) - F(2), where the numbers inside the brackets indicate the topology of each layer.
The authors of [5], [6] have tried to gain insight into the operation of the CNN models via visualizing the features and filters of their hidden units. The authors of [6] have shown that the CNN model is highly sensitive to local structure of the input image and that particular regions of an image are responsible for firing specific neural units. [5] discovers that there exist many invariant detectors for faces, shoulders, texts, etc in convolutional layers. It shows that CNNs can learn partial information even though no explicitly labeled faces or texts exist in the training datasets.

If we can be certain that specific objects will fire specific hidden neurons in a CNN, then it will be very useful because we can infer the objects in the input image from the response of the hidden neurons. Based on above observations, we believe it is possible to fine-tune a CNN such that particular neurons will respond to specific objects if suitably prepared object specific training examples are used. To verify this idea, we use Annotated Facial Landmarks in the Wild (AFLW) database [24] as training dataset to fine-tune the AlexNet. AFLW database contains 25,993 faces in 21,997 real world images collected from flickr with a range of diversity and variation in poses, ages and illuminations. Unlike most common methods that crop the ground-truth face area as positive samples and non-face area as negative samples, we use the original images as positive samples. For negative samples, we also use the same images, but mask the facial parts with random noises (R, G, B value are randomly generated from 0-255). Some examples of these positive and negative samples are shown in Fig. 2. The idea is that if the CNN model can discriminate the masked image from the original image, it will be forced to use the partial information from face area for classification. What we desire is that through re-enforcing the CNN to discriminate images with faces and images without faces, then the network could organise itself such that certain units will be responsible for representing faces, thus enabling the extraction of face specific channels for various post-processing.

3.2.2 Face Specific Convolutional Channel Identification

In order to quantify how well each convolutional channel responds to faces after fine-tuning, we’ve studied the 5th convolutional layer which has a 13 x 13 x 256 topology and can be visualized as 256 different 2-D heatmaps (13 x 13). We resize every heatmap from shape (13 x 13) to (227 x 227), the same size as the input image, using a bicubic interpolation. We calculate the average intensity value of the heatmap both inside and outside the face areas respectively, which are denoted as “face-score”, i.e., \( \frac{1}{w \times h} \sum_{i=1}^{w} \sum_{j=1}^{h} I_{i,j} \), for a given bounding box, where (x, y) is the top left coordinate, (w, h) are the width and height, and \( I_{i,j} \) is the intensity value at point (i, j). The resized heatmap has the same shape as the input image, and the face area in the heatmap can be located by directly using the ground-truth face annotations. To calculate the face-score in the face areas, all the intensity value of pixels inside the face area are added up and divided by the number of pixels in that face area. Similarly, the face-score outside the face area can be calculated easily. We then use 1,000 images randomly chosen from AFLW [24] to calculate the face-scores inside and outside face areas of all the channels in the 5th convolutional layer of the fine-tuned model. The final value of face-score is the average value across all the 1,000 images. The results are shown in Fig. 3. We can see that the 196th channel has the highest face-score inside the face areas, followed by the 139th channel. The value of face-score outside face area is small in all channels. This shows that there do exist face specific channels where specific neurons are fired at the spatial positions corresponding to the face regions in the input image. Though it seems empirical to choose the channel index, however consistent responding channels can be produced by randomly choosing different face images. We use the 196th channel in all experiments, which can get better results than using 139th channel.

3.2.3 Multi-Scale Features

From above, we could compute the face heatmap (conv5196) and generate the bounding box of potential face regions by a given threshold. As a result, the bounding box could be used to represent the face position. However, as shown in Fig. 4, there are two problems if we want to use the heatmap directly for face detection. One is that the heatmap cannot capture small faces, the other is that the strong firing neurons in the heatmap tend to shift to the edge of the image if the face is not in the center of an image. We have developed a multi-scale approach to solving these problems: The input image is divided into small sub-images by a sliding window at multiple scales. We specify the stride, with which the sliding window is moved each time, to be half of the sub-image’s size. Thus adjacent sub-images are able to overlap with each other. In this work, the sizes of the sub-images are 1/4, 1/6 and 1/8 of the size of the original image to achieve multi-scale feature extraction. All sub-images and the original image are then resized and fed to the fine-tuned CNN to extract the channel heatmaps (conv5196), which are then merged to be a single heatmap according to the correspond-
Fig. 4 Examples for single-scale and multi-scale feature extraction.

The merging is achieved by selecting the maximum intensity value of each heatmap at each pixel position in the original image scale, i.e., $I_{i,j} = \max_{1 \leq l \leq n} I_{i,j}^l$, where $I_{i,j}$ is the intensity value of the merged heatmap at point $(i, j)$ in the original image scale, and $I_{i,j}^l$ is the intensity value of the $l^{th}$ sub-image’s heatmap at point $(i, j)$. As a result (see Fig. 4), the merged heatmap is able to capture small faces and locate faces in the input image more precisely. This is because small faces in the original scale become “larger” in the sub-image scale, which can be captured by the fine-tuned CNN.

3.2.4 Fast Non-Face Region Filtering

The face heatmap contains information about the likelihood of a pixel belonging to a face region. The higher the intensity the more likely it is from a face region. Therefore, this allows us to quickly filter out non-face regions by simply setting a threshold to the heatmap. Only pixels above a certain threshold will likely contain faces. Therefore, in the face detection stage, we only need to search regions covered by these pixels thus significantly reducing the search space and speeding up face detection. To get an idea about the kind of saving we can achieve, we randomly pick 1,000 images from ALFW and threshold the heatmaps. We count the average number of pixels above the threshold which contains areas of potential faces. At the same time, we also count the average number of faces that are covered by these remaining pixels. Obviously, we want the number of remaining pixels to be small and at the same time these remaining pixels should cover all genuine faces. Figure 5 shows the percentage of pixels above different thresholds, and for each threshold, the percentage of genuine faces covered by the remaining pixels. It is seen that by simply thresholding the heatmap, we can dramatically reduce the search regions without missing genuine faces. For example, by setting a threshold value to about 110, just a little over 10% of the pixels remain and at the same time all faces are covered by these pixels. This means that compared with full search, this procedure will reduce face detection search region by 90%.

4. Face Proposals And Detection

The second stage of our method (Fig. 1 (b)) locates faces more precisely and separates overlapped faces which cannot be distinguished in the heatmap. We first use face-score defined above (average intensity of a specific area in a heatmap) to quickly select face proposals. Then another binary CNN classifier is trained to discriminate face images from non-face images. In the end, Non-Maximum Suppression is used to merge multiple face windows for final detection.

4.1 Face Proposals by Face-Score

Instead of using category-independent region proposals method like selective search [26] used in R-CNN[2] for general object detection, we simply use a multi-scale sliding window to select potential face windows based on face-score defined above. Specifically, while scanning the input image, the face-score of the sliding window is calculated at the same position in the corresponding heatmap. The sliding windows are selected as face region proposals if the face-scores exceed a given threshold (80 in our case). This approach can reject non-face regions effectively based on the above observation that the pixel intensity values around the face regions are higher than non-face regions. All potential face proposals of one image are collected as a batch to feed to a binary CNN classifier described below.

4.2 Candidate Face Window Selection by CNN

Due to a range of visual variations such as poses, expressions, lightings and occlusions, a robust face classifier is trained by using augmented data. The fine-tuned CNN
model used to extract face response heatmap is used as pre-trained model, and then fine-tuned again with face images and non-face images. All the face images are cropped from AFLW dataset [24] by the ground-truth bounding box, and augmented by making them darkened, blurred and occluded (Fig. 6(a)). Non-face images are collected in the following ways (Fig. 6(b)): (1) background images randomly cropped from AFLW images with a given Intersection-over-Union (IoU) ratio (0, 0.1 and 0.2) to a ground-truth face; (2) cropped by double-sized ground-truth faces; (3) face im-

---

Fig. 6 Example face images (a) and non-face images (b) from AFLW for face classifier training.

---

Fig. 7 Performance comparison on PASCAL Face dataset for the original (left) and adjusted annotations (right).

---

Fig. 8 Performance comparison on FDDB dataset with discrete and continuous protocols.
Fig. 7 Qualitative face detection results and face response heatmap on PASCAL face (top three rows) and FDDB dataset (bottom three rows).

ages padded with non-face images. Each candidate face window has a classification score after being processed by the trained binary classifier. Finally all detected face windows are sorted from highest to lowest based on classification score, and then non-maximum suppression (NMS) is applied to the detected windows to reject the window if it has an Intersection-over-Union (IoU) ratio bigger than a given threshold.

4.3 Face Detection Experiments

As described above, the AFLW dataset is used to train our model, and then we use PASCAL Face [13] and FDDB dataset [27] to evaluate our face detector. PASCAL Face dataset is a widely used face detection benchmark, containing 851 images and 1,341 annotated faces. FDDB dataset is a larger face detection benchmark, consisting of 5,171 annotated faces in 2,845 images. It contains a wide range of difficulties including occlusions, various poses and out-of-focus faces.

One problem in the evaluation of face detection is the different annotations between training datasets and testing datasets such as policies for what constitutes a face, size of annotation boxes and minimum/maximum of face size. In order to solve this problem, some works [11], [14], [21] try to manually adjust the annotations to get better results. In our work, we use both the original and adjusted annotations [14] of PASCAL Face dataset with the toolbox provided by [14]. FDDB dataset is evaluated with the original elliptical annotations by two evaluation protocols provided in [27]: the continuous score and discontinuous score. In order to better fit the elliptical annotations that cover the whole faces, we extend our detected square boxes vertically by 40% to upright rectangles for FDDB dataset. We also fit the largest upright ellipses for the extended rectangles as elliptical outputs for evaluation (Fig. 9).

We report the average precision on PASCAL Face dataset (Fig. 7), discontinuous and continuous ROC for all the 10 folds (Fig. 8 (a) and (b)) and individual ROC (Fig. 8 (c) and (d)) for each fold on FDDB dataset. By comparing with other state-of-the-art face detectors [9], [15], [21], [22], [28]–[31], our method can achieve similar results both on PASCAL faces [13] and FDDB [27] datasets while having lower complexity. The runtime of our model is around 70 ms on a single GPU (Nvidia Tesla k40) for a 640 × 480 VGA image.

In addition, we design several comparative experiments to investigate the impact of data augmentation. The CNN bi-
nary classifier in the second stage is retrained with a slightly different training dataset, while all the other components are left unchanged. For face images, we remove the darkened, blurred and occluded face images (Fig. 6 (a)) respectively, and then the padded and double-sized images (Fig. 6 (b)) are removed from non-face images. As before, all the newly trained models are tested on FDDB dataset as shown in Fig. 8 (e) and (f). We can see that the augmented training dataset has a significant influence on the performance of the detector. By removing one type of augmented training data, there is an obvious drop in the performance of detection. For example, the padded and double-sized non-face images could help the model locate face area more precisely, i.e., neither too big nor too small, thus bounding box regression is not needed for post-processing. Therefore, better data augmentation makes full use of the high-capacity of CNN to achieve better performance. In our case, all kinds of facial variations can be represented in a single CNN model.

5. Discussion

The contribution of this paper is not only on the face detector itself, but rather the introduction of a face specific channel in a deep neural network that can be easily exploited for face detection. This is the first work that introduces the concept of object specific channel in a deep convolutional neural network. Although we use face as a specific case study, our method does not depend on any human knowledge about faces or any special processing for faces. Our method could be extended to other classes of objects in a similar way to turn the convolutional feature maps of a CNN into object specific feature channels for object detection and other computer vision tasks.

Compared to many methods in the literature, our method is much simpler: ground-truth bounding boxes are the only information needed to train our model, and one single model can capture all the facial variations based on the carefully designed data augmentation. In contrast, Face-ness [18] needs additional hair, eye, nose, mouth and beard annotations to train several attribute-aware face models and uses bounding box regression to refine detected windows. DPM and HeadHunter [14] use extra annotation to train view-specific components to tackle facial variations. Joint-Cascade [29] uses face alignment to help face detection with manually labeled 27 facial points. Some qualitative results on PASCAL Face and FDDB dataset together with the face response heatmaps are shown in Fig. 9. It is seen that our detectors can successfully detect faces in challenging settings.

6. Concluding Remarks and Future Work

In this paper, we have developed a method to exploit the internal representation power of hidden units of a deep convolutional neural network. We seek to discover and exploit the convolutional channels of a CNN in which neurons are activated by the presence of specific objects in the input image. Through a purposefully designed face specific training, object specific channels are extracted and automatically identified for human faces. Building on the object specific channels, a multi-scale approach are used to build state of the art face detectors with the advantage of being simple and compact.

Acknowledgements

The author acknowledges the financial support from the International Doctoral Innovation Center, Ningbo Education Bureau, Ningbo Science and Technology Bureau, and the University of Nottingham. This work was also supported by the UK Engineering and Physical Sciences Research Council [grant number EP/L015463/1].

References

HOU et al.: OBJECT SPECIFIC DEEP FEATURE FOR FACE DETECTION


