Object Specific Deep Feature and Its Application to Face Detection

Xianxu Hou
University of Nottingham
Ningbo China
xianxu.hou@nottingham.edu.cn

Jiasong Zhu
Shenzhen University
Shenzhen China
zhujiasong@gmail.com

Ke Sun
University of Nottingham
Ningbo China
ke.sun@nottingham.edu.cn

Linlin Shen
Shenzhen University
Shenzhen China
llshen@szu.edu.cn

Guoping Qiu
University of Nottingham
Ningbo China
guoping.qiu@nottingham.ac.uk

Abstract

We present a method for exploiting object specific deep features and use face detection as a case study. 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 an object specific channel (OSC) and systematically identifying it for the human face object has been developed. Building on the basic OSC features, we introduce a multi-scale approach to constructing robust face heatmaps for rapidly filtering out non-face regions thus significantly improving search efficiency for face detection in unconstrained settings. We show that multi-scale OSC can be used to develop simple and compact face detectors with state of the art performance.

1 Introduction

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 [14]. 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 Figure 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).

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.

2 Related Work

Face detection models in the literature can be divided into three categories: Cascade-based model, Deformable Part Models (DPM)-based model and Neural-Network-based model. The most famous cascade-base model is the VJ detector [10] based on Haar-like features. Some works try to improve VJ-detector via using more complicated features such as SURF [15]. Another category is DPM-base model [2], which treats face as a collection of small parts. Neural-Networks-based detectors are usually based on deep convolutional neural networks. Faceness [13] 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 [7]. The CCF detector [12] uses an integrated method called Convolutional Channel Features. Cascade architectures based on CNNs [6] have been also designed to help reject background regions at low resolution, and select face area carefully at high resolution.

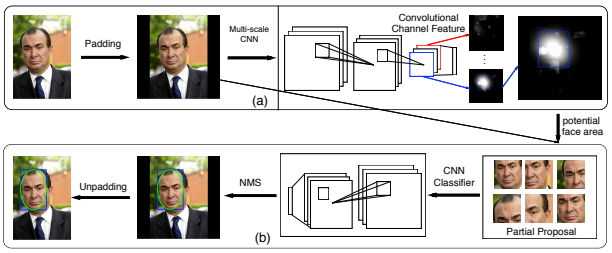


Figure 1. Model Overview

3 Method Overview

Our goal is to discover and exploit face specific convolutional channel to help locate the face areas quickly for further processing. Our system contains two stages as shown in Figure 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) [9] to obtain the final detection results.

Figure 2. Examples of masked face images (first row) and original images with faces (second row).

3.1 Training Data and CNN Fine-tuning

We use Annotated Facial Landmarks in the Wild (AFLW) database [4] as training dataset to fine-tune the AlexNet [5]. 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 Figure 2. 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 Face Specific Convolutional Channel

In order to quantify how well each convolutional channel responds to faces after fine-tuning, we've studied the 5^{th} 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., $\frac{1}{wh}\sum_{i=x}^{x+w}\sum_{j=y}^{y+h} I(i,j)$, 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. We use 1,000 images randomly chosen from AFLW [4] to calculate the face-scores inside and outside face areas of all the channels in the 5^{th} 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 Figure 3. We can see that the 196^{th} channel has the highest face-score inside the face areas, followed by the 139^{th} 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. We use the 196^{th} channel in all experiments, which can get better results than using 139^{th} channel.

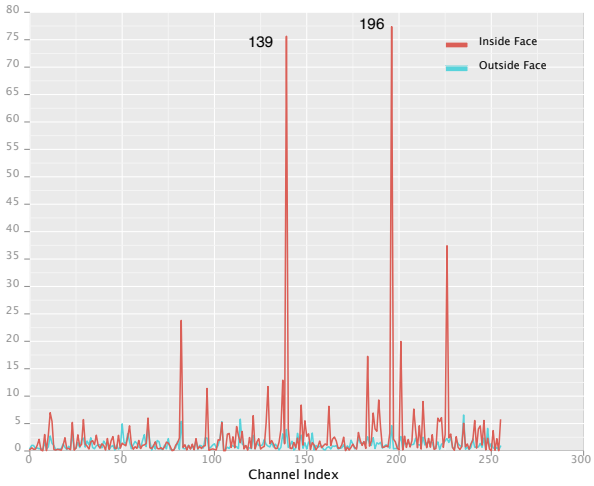


Figure 3. Face-scores of inside and outside face area for each channel in $conv5$ layer, averaged over 1,000 images randomly chosen from ALFW.

3.3 Multi-scale Features

As shown in Figure 4, the heatmap cannot capture small faces, we have developed a multi-scale approach to solving this problem: 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. All sub-images and the original image are then resized and fed to the fine-tuned CNN to extract the channel heatmaps ($conv5_{196}$), which are then merged to be a single heatmap according to the corresponding locations in the original image scale. Specifically, each heatmap of sub-images are extended to the original image scale, and the intensity value outside the sub-image areas are set to be zero. The merging is achieved by selecting the maximum intensity value of each heatmap at each pixel position in the original image scale, i.e., $I_{i,j} = \max_{1\le l\le n} I^l_{i,j}$, where $I_{i,j}$ is the intensity value of the merged heatmap at point (i, j) in the original image scale, and $I^l_{i,j}$ is the intensity value of the l^{th} sub-image's heatmap at point (i, j). As a result (see Figure 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.

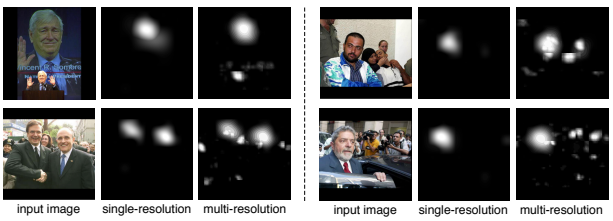


Figure 4. Examples for single-scale and multi-scale feature extraction.

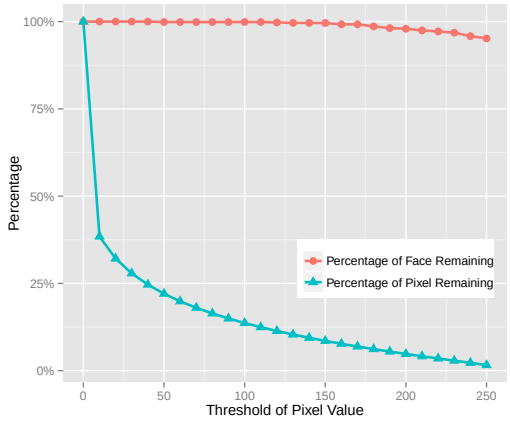


Figure 5. Diagram for the average percentage of remaining pixels (green line) and remaining faces (red line) above different thresholds.

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

3.4 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.

4 Face Proposals And Detection

4.1 Face Proposals by Face-Score

We 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

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 [4] by the ground-truth bounding box, and augmented by making them darkened, blurred and occluded (Figure 6(a)). Non-face images are collected in the following ways (Figure 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 images 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.

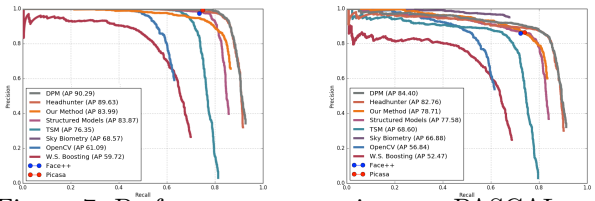
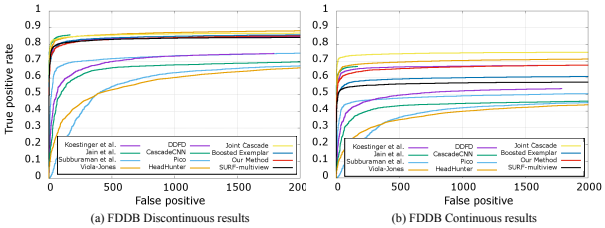


Figure 7. Performance comparison on PASCAL Face dataset

4.3 Face Detection Experiments

As described above, the AFLW dataset is used to train our model, and then we use PASCAL Face [11] and FDDB dataset [3] 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



(a) FDDB Discontinuous results (b) FDDB Continuous results

Figure 8. Performance comparison on FDDB dataset with discrete and continuous protocols.

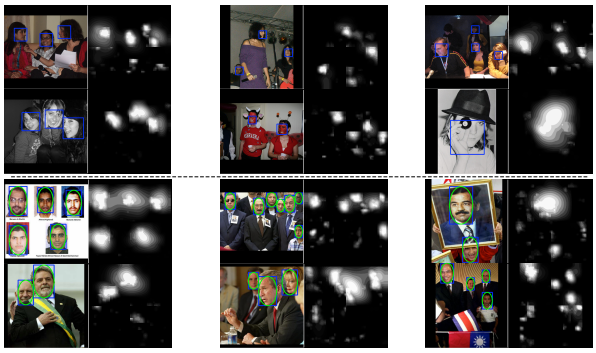


Figure 9. Qualitative face detection results and heatmap on PASCAL face (top three rows) and FDDB dataset (bottom three rows).

2,845 images. It contains a wide range of difficulties including occlusions, various poses and out-of-focus faces.

In our work, we use the original and adjusted annotations [8] of PASCAL Face dataset with the toolbox provided by [8]. FDDB dataset is evaluated with the original elliptical annotations by two evaluation protocols provided in [3]: 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 (Figure 9). We report the average precision on PASCAL Face dataset (Figure 7), discontinuous and continuous ROC for all the 10 folds (Figure 8 (a) and (b)).

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, Faceness [13] 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 [8] use extra annotation to train view-specific components to tackle facial variations. JointCascade [1] 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 Figure 9. It is seen that our detectors can successfully detect faces in challenging settings.

5 Concluding Remarks

In this paper, we have developed a method to exploit the internal representation power of hidden units of a deep neural network. We explicitly set out to seek convolutional channels that specifically respond to face areas in images. Through a purposefully designed face specific training samples, we show that we can fine-tune a pretrained CNN to reinforce the internal face specific convolutional channels, which can be used to build face detectors with the advantage of being simple and compact. Our method could be extended to other objects, and we are currently applying it to objects from cars in photography images to cells in medical images.

References

[1] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *Computer Vision–ECCV 2014*, pages 109–122. Springer, 2014.

[2] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[3] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

[4] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 2144–2151. IEEE, 2011.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5325–5334, 2015.

[7] J. Liu, Y. Deng, and C. Huang. Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv preprint arXiv:1506.07310*, 2015.

[8] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *Computer Vision–ECCV 2014*, pages 720–735. Springer, 2014.

[9] A. Neubeck and L. Van Gool. Efficient non-maximum suppression. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 850–855. IEEE, 2006.

[10] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[11] J. Yan, X. Zhang, Z. Lei, and S. Z. Li. Face detection by structural models. *Image and Vision Computing*, 32(10):790–799, 2014.

[12] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Convolutional channel features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 82–90, 2015.

[13] S. Yang, P. Luo, C.-C. Loy, and X. Tang. From facial parts responses to face detection: A deep learning approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3676–3684, 2015.

[14] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[15] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1491–1498. IEEE, 2006.