Improving Face Image Extraction by Using Deep Learning Technique

Zhiyun Xue, Sameer Antani, L. Rodney Long, Dina Demner-Fushman, George R. Thoma
National Library of Medicine, NIH, Bethesda, MD

ABSTRACT

The National Library of Medicine (NLM) has made a collection of over a 1.2 million research articles containing 3.2 million figure images searchable using the Open-i[SM] multimodal (text+image) search engine. Many images are visible light photographs, some of which are images containing faces ("face images"). Some of these face images are acquired in unconstrained settings, while others are studio photos. To extract the face regions in the images, we first applied one of the most widely-used face detectors, a pre-trained Viola-Jones detector implemented in Matlab and OpenCV. The Viola-Jones detector was trained for unconstrained face image detection, but the results for the NLM database included many false positives, which resulted in a very low precision. To improve this performance, we applied a deep learning technique, which reduced the number of false positives and as a result, the detection precision was improved significantly. (For example, the classification accuracy for identifying whether the face regions output by this Viola-Jones detector are true positives or not in a test set is about 96%.) By combining these two techniques (Viola-Jones and deep learning) we were able to increase the system precision considerably, while avoiding the need to manually construct a large training set by manual delineation of the face regions.

Keywords: image modality classification, deep learning, convolutional neural networks, face detection, Viola-Jones algorithm

1. INTRODUCTION

The U.S. National Library of Medicine (NLM) has developed a multimodal (text + image) biomedical search engine called Open-i[SM1]. It provides capability to search figures in biomedical scientific publications, using both text query and/or image query. Currently, Open-i provides open access to nearly 3.2 million images from approximately 1.2 million Open Access biomedical research articles obtained from the NLM's PubMed Central (PMC) repository. Open-i also provides various filters to limit the search space. One such filter is *image type* (*modality*). Besides medical modalities, such as MRI, CT, X-ray, and ultrasound, there is also a category of *photograph* (visible light image). The *photograph* category contains many images, clinical and non-clinical, and the content can be very diverse. For further classification of the *photograph* category, we have been developing methods for extracting images from photographs in categories such as skin tissue and endoscopic images [1, 2]. In this paper, we report on the extension of our work to extracting *face images* (images containing faces) from photographs.

The goal in face detection is to identify and isolate human faces visible in a photographic image. Reliable face detection is one of the most studied research topics in the field of computer vision and precursor to face identification or matching. For a good survey paper on this topic, see [3]. The Viola-Jones detector is a multi-stage classification framework that was first proposed by Paul Viola and Michael Jones in 2001 [4]. It may be the most commonly used method for face detection, although it can also be trained to detect various other objects. An implementation of this algorithm which has been trained with another face image dataset is provided in OpenCV (and Matlab) (detailed information on the implementation and the face image data used is provided in Section 2.1). We tested the pre-trained detector first on the Face Detection Data Set and Benchmark (FDDB) dataset designed for studying the problem of unconstrained face detection [5]. This data set contains 2845 images collected from news photographs containing 5171 faces. The Viola-Jones detector obtained very high precision (approximately 0.94) for the FDDB dataset. (We present the results in detail in Section 2). For our application, high precision (which means most of the extracted images are truly images containing faces) is more desirable than high recall. We applied the Viola-Jones algorithm to our dataset that contains 115,370 photographs from Open-i, with the result that the face images extracted include many false positives, and the detection

[1] http://Open-i.nlm.nih.gov

precision is low. This may be explained by that the Viola-Jones detector implemented in OpenCV was possibly trained by using positive samples similar to the images in the FDDB. However, for our Open-i dataset where the images are figures from published literature is quite different from the FDDB dataset. It may, in fact, also be different from several other publicly available face detection image datasets that have often been used and tested, as reported in the face detection literature [6, 7]. According to our analysis, the differences include the following aspects:

- The content of images in the Open-i dataset is much more diverse.
  The figures in the *photograph* category include all kinds of images that are excluded from the specific categories, such as CT, MRI, X-ray, ultrasound, and graphical figures, charts, and tables. For example, the photograph category may include, but is not limited to, images of various instruments, body parts, animals, plants, chemicals, scenes, illustrations, and maps. Even for photographs with faces, the Open-i dataset contains faces that may have abnormalities, are de-identified for privacy (black strip across the eyes), have annotations (sketch marks made prior to surgery, or for medical measurements), paintings/drawings of faces, and so on. Figure 1 shows some examples of face images in this dataset.

- Open-i dataset contains many non-face images in addition to face images.
  Unlike the FDDB dataset in which there is at least one face region in every image, the majority of Open-i photographs in our collection do not contain any faces. In addition, the content in these non-face images is also extensively wide-ranging as stated previously. Figure 2 shows several non-face image examples in the Open-i dataset.

- The number of images in the Open-i dataset is much larger.
  Currently there are 115,370 Open-i photographs in the dataset we are using.

Therefore, the challenges of extracting face images from a biomedical photograph dataset, such as the Open-i collection, include not only the many variations in scale, location, orientation, pose, facial expression, lighting conditions, and occlusions in the face images as pointed out by the recent survey on face detection [3], but also the large variance in image content and image quality/resolution in both non-face images and face images. These challenges influence the performance of algorithms with respect to both precision and recall. We could retrain the Viola-Jones detector using the positive and negative samples from our dataset to see if the performance can be improved. However, it requires a lot of manual labor to label images (marking face regions) in such a huge dataset. Instead we tried a different approach, aiming to refine the results obtained by the pre-trained Viola-Jones face detector and improve the performance with respect to precision at first.

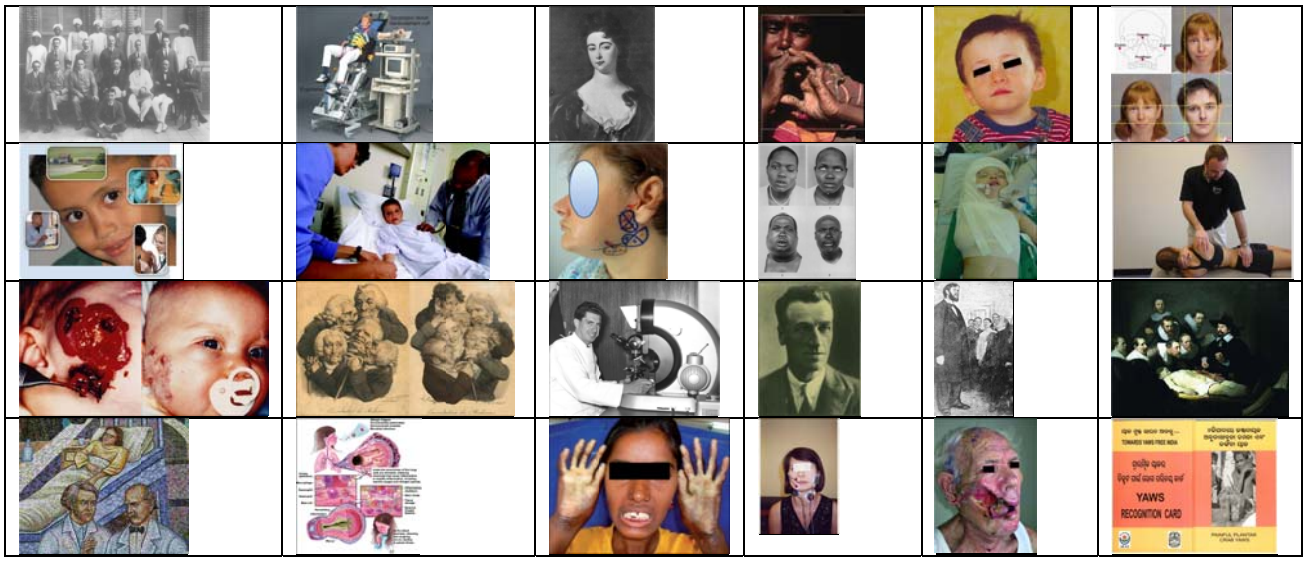

Figure 1. Examples of face images in Open-i

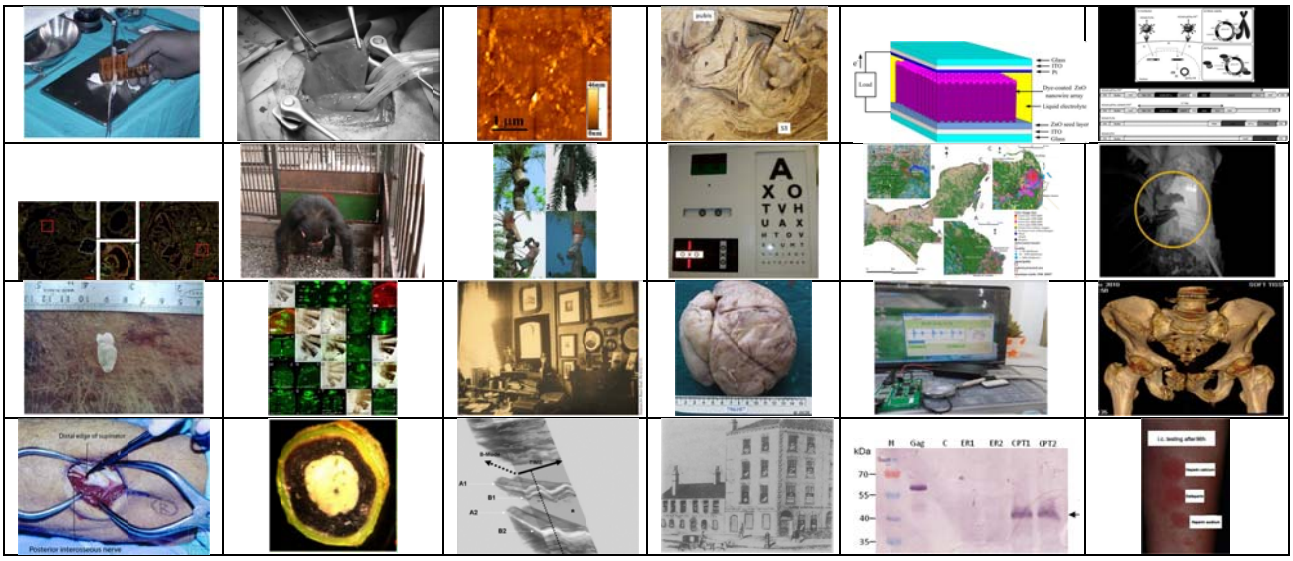

Figure 2. Examples of non-face images in Open-i

In recent years, the technique of deep learning has attracted the attention of researchers in both academia and industrial fields because of its state-of-the-art performance in applications of computer vision and speech recognition. Deep learning is feature/representation learning which can automatically discover multiple levels of representations from raw input data [8, 9]. Compared to conventional methods which are based on handcrafted feature extractors, deep learning avoids the process of handcrafting (which requires significant engineering skills and domain knowledge) to automatically obtain good features through a learning procedure. Deep learning methods employ architecture with multiple layers, in which the higher the layer, the more abstract representation the learning yields. The convolutional neural network (CNN) is one particular type of deep neural network [10]. It has become the leading deep learning method used for image classification and object recognition since the ImageNet [11] competition in 2012, in which it achieved the best result, substantially outperforming the competition. In this paper, we used CNN to refine the result of Viola-Jones face detection. Specifically, we examined all those face region candidates extracted by the Viola-Jones detector and identified false positives and true positives. We then used this dataset of candidate face regions to train and test the convolutional neural network to separate false positives from true positives. Therefore, in our approach, the Viola-Jones detector (trained using a different face dataset) is used for face localization and detection, and CNN is used for reducing the number of false positives and improving the precision. By combining these two techniques this way, we avoided the intense labor work for manual delineation of face regions in images of a large and diverse dataset and improved the precision of the system significantly.

The rest of the paper is organized as follows. Section 2 describes and analyzes the testing results of the Viola-Jones (VJ) face detector on both FDDB and Open-i datasets. Section 3 describes using CNN to improve the results of the VJ detector and discusses the results. The conclusions and future work is given in Section 4.

2. EXTRACTION OF FACE REGION CANDIDATES

2.1 Viola-Jones face detector

The VJ algorithm [4] is a well-known method for detecting faces and has a significant impact in the face detection field. The VJ algorithm consists of four major components: 1) a set of rectangle features which are reminiscent of Haar basis functions; 2) the integral image; 3) a classifier built on the AdaBoost learning algorithm; and 4) the combination of classifiers in cascade architecture.

The Haar-like rectangle features in [4] include three kinds of features as shown in Figure 3: a two-rectangle feature that calculates the difference between the sum of the pixels within two rectangles, a three-rectangle feature that calculates the

difference between the sum within two outside rectangles and the sum within the central rectangle, and a four-rectangle feature that calculates the difference between the sums within the diagonal pairs of rectangles.

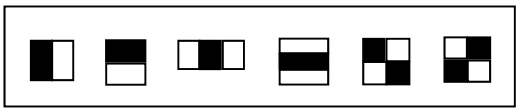

Figure 3. Haar-like rectangle features

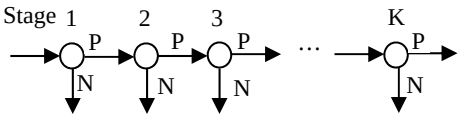

Figure 4. Cascade of classifiers

The number of rectangle features that need to be calculated for an image (at many scales) is very large. To make the computation of rectangle features fast, Viola and Jones applied the integral image. The integral image is a summed area table in which the value at the location (x, y) is the sum of all the pixels above and to the left of (x, y). The integral image can be computed efficiently in just one pass over the original image. The Haar-like rectangle features, which are based on the sum of pixel values in rectangles, can be calculated rapidly by using the computed integral image [4].

Each basic classifier is trained using the AdaBoost learning algorithm [12]. AdaBoost (Adaptive Boosting) is a boosting algorithm which combines a set of weak classification functions (weak learners) to produce one stronger classifier. In [4], each of the weak learners depends on a single feature and determines the optimal threshold classification function for that feature. Therefore, the AdaBoost can also be used as a feature selector by identifying key weak learners.

A cascade of those basic classifiers is then constructed. A positive result from a classifier will be passed to the next classifier while a negative outcome is rejected immediately, as illustrated in Figure 4. Each classifier is trained to have a very high detection rate (trying to keep all the positives while rejecting a certain amount of negatives) and uses the samples that pass through all the previous stages to train. Thus subsequent classifiers face harder tasks and usually have more key weak learners (or features) selected. The cascade attempts to make background regions (negatives) to be discarded quickly, at the earliest possible stage, based on the observation that the majority of the sub-windows in an image are non-face regions.

After training the VJ detector with positive and negative samples of faces, given a test image, the detector scans across the image at multiple scales and locations to find the sub-windows that contain a face. The detector also combines overlapping multiple detections of one face into a single detection of the face in the post-processing stage. Although the training of the VJ detector may be slow, the detection is very fast and can be used for real-time processing.

In follow-up work, researchers have tried to improve the performance of the Viola-Jones face detector with respect to the features and learning algorithm. Features such as joint Haar-like features [13], anisotropic Gaussian filters [14], local binary patterns (LBP) [15], and histogram of oriented gradients (HOG) [16], have been proposed to address the limitations of the original set of rectangle features. The learning algorithms, such as variations of boosting learning algorithms, classification and regression tree (CART) [17], support vector machine (SVM) [18], and neural networks [19], have been applied to replace the standard AdaBoost algorithm used in [4].

For our experiments, we applied the implementation provided by the Matlab Computer Vision Toolbox [20] (it calls the corresponding OpenCV face detector which were developed by Lienhart et. al [17]). This MATLAB (and OpenCV) algorithm is an improved version of the original VJ face detector. Specifically, there are two extensions to the original VJ algorithm. One is that a new set of rotated Haar-like features have been added. The other is that Gentle AdaBoost [21] with small CART trees are used as base classifiers. The images used to train the detector appear to have been from the Facial Recognition Technology Database (FERET), although the relevant publications [22, 23] are not explicit on this point. We will refer to this detector/training set as the Viola-Jones (or VJ) FERET-trained detector. The fully trained cascade consisted of 20 stages. Each stage used 5000 positive and 3000 negative face region samples filtered by previous stages to train. The 5000 positive samples were generated from 1000 original face regions by random rotation, scaling, mirroring and shifting. The classifier at each stage was trained to detect 99.9% of face samples while rejecting half of the non-face samples (i.e., with the performance of false positive rate being 0.5 and true positive rate being 0.999). For the details of the algorithm and the process of training, please refer to [17]. There are several important parameters in the

implementation [20]: 1) the size of the smallest face to detect; 2) the size of the largest face to detect; 3) the scale factor that incrementally scales the detection resolution between the minimum and the maximum size of the face object to detect; and 4) the threshold for the number of times a target object needed to be detected during the multiscale detection phase. We tested the trained VJ face detector on both the FDDB dataset and the Open-i dataset. For both tests, the default values of those parameters were used.

2.2 Testing on the FDDB dataset

The FDDB dataset [5] is a benchmark intended for use in evaluating face-detection algorithms in unconstrained settings. It contains 2845 grayscale and color images that are selected from a dataset [24] collected from news articles. Each of the 2845 images contains at least one face. The dataset also includes annotations of a total of 5171 face regions identified by manual annotation. The FDDB dataset contains a wide range of different faces, including faces with occlusions, low resolution, out-of-focus, and difficult poses. The reason we selected the FDDB dataset to test the VJ detector is because of the unconstrained settings and wide range of different faces represented, two characteristics that the Open-i dataset also exhibits. The ground truth face regions provided by the FDDB dataset are elliptical regions. The matching rule we used is: if the center of the extracted box is inside the ground truth elliptical region, it is considered being true positive; otherwise it is a false positive. As described previously, we used the Matlab implementation of the detection algorithm; for parameter choices, we used default values. The results are listed in Table I. The number of face regions detected by the VJ detector was 4131. Among them, 3902 regions are true positive while 299 regions are false positives. Therefore, the precision (the ratio of the number of true positives and the number of extracted regions) is 0.945 and the recall (the ratio of the number of true positives and the number of ground truth regions) is 0.755. The recall is relatively moderate but the precision, very high. As we stated previously, high precision is more desirable for our application if there is a trade-off between recall and precision. We then applied the same detector to our Open-i dataset.

Table I. Detection results of the Viola-Jones detector on the FDDB dataset

Ground Truth	Extracted	True Positive	False Positive	False Negative	Precision	Recall
5171	4131	3902	299	1269	0.945	0.755

2.3 Testing on the Open-i dataset

Out of 115,370 photographs, the VJ detector extracted 20,400 face images which contain 30,390 face regions. Note that face image extraction, which is to identify an image that contains a face, is less strict than face detection. For example, for an image having one or multiple faces, the image can be correctly labeled as a face image even if a region is incorrectly identified as a true face region. To evaluate the performance of the VJ detector for face region detection, we manually labeled each box extracted by the detector as a true positive or a false positive. If an image contains at least one true positive, then it is a true face image; otherwise, it is not. Table II lists the results for face region extraction on the Open-i dataset. Out of 30,390 extracted face regions, only 9,357 of them are true positives. As a result, the precision for face region extraction is very low, around 0.31. The number of images containing those true face regions is 5,208. Because we do not have ground truth for the entire Open-i dataset, we cannot calculate the number of false negatives and the value of recall. Figure 5 shows some true positives obtained by the VJ detector while Figure 6 shows some false positives. The identified face areas are marked by yellow boxes with a label of “Face”.

Table II. Face region detection results of the Viola-Jones detector on the Open-i dataset

Extracted Regions	True Positive	False Positive	Precision
30,390	9,357	21,033	0.31

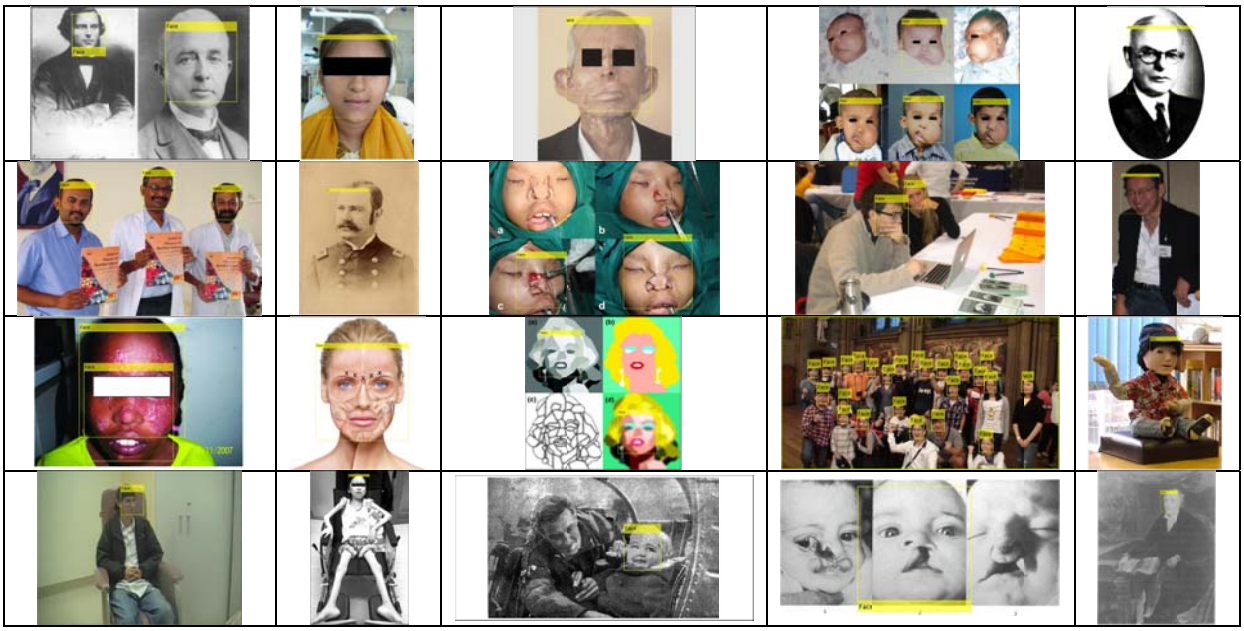

Figure 5. True positives obtained by the Viola-Jones face detector

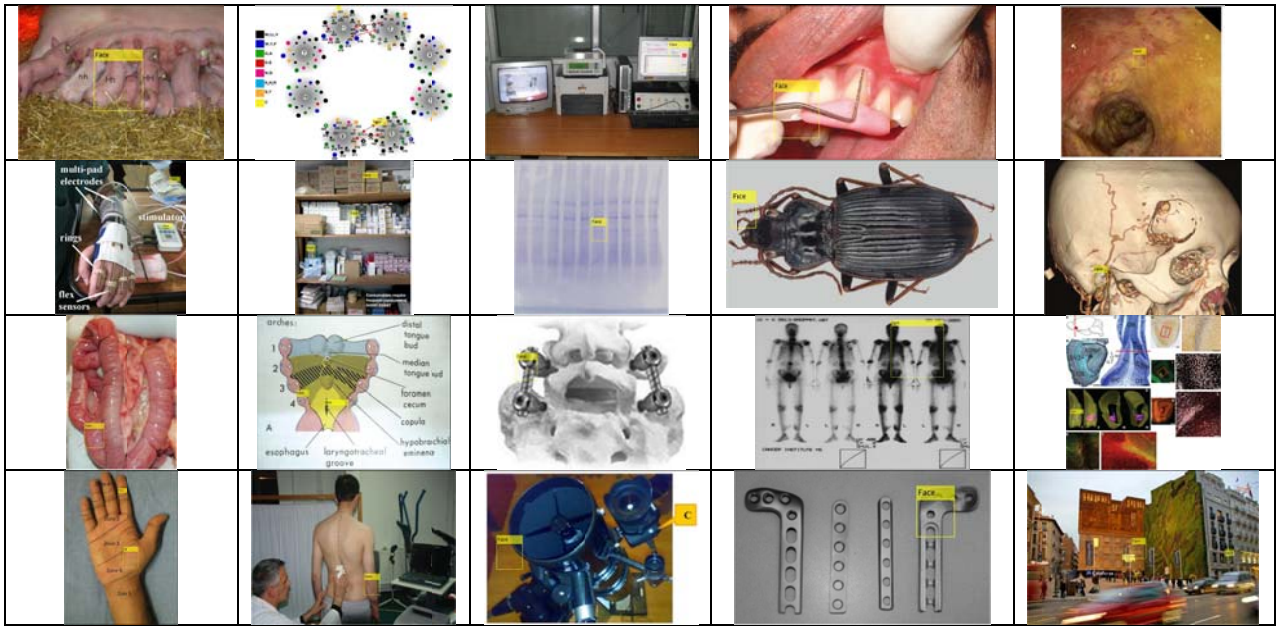

Figure 6. False positives obtained by the Viola-Jones face detector

3. IMPROVEMENT USING CONVOLUTIONAL NEURAL NETWORKS

Because of the high false positive rate in the extracted face region candidates that we observed when applying the Viola-Jones FERET-trained detector, we then trained a convolutional neural network to classify those face region candidates into two classes: face region and non-face region. In the following, we first introduce the technique of convolutional neural networks and then discuss the results. We also retrained the Viola-Jones face detector using these candidate face regions and compared with the results of the convolutional neural network.

3.1 Convolutional Neural Networks

The convolutional neural network (CNN) was first introduced in the 1990s, by LeCun, et.al. [25], for the task of classifying images of handwritten digits. Interest in convolutional neural networks was reignited in the computer vision field after a large, deep CNN was used to classify 1.2 million high-resolution images into 1000 classes in the ImageNet Large-Scale Visual Recognition Challenge (LSVRC)-2012 contest and achieved considerably better performance than other state-of-the-art methods [26]. CNNs are a special kind of multi-layer neural network designed to process images. They explore spatial relationships of pixels in images to reduce the number of parameters in the neural network that must be trained. There are four important ideas used by CNNs:

1) Local connections: each unit in one layer is connected to a spatially-connected local subset of units in the previous layer;
2) Shared weights: all units in each of the feature maps in one layer have the same set of weights;
3) Pooling: a spatial subsampling step is applied to reduce the dimensions of the feature maps;
4) Many layers: the network may have more than 10 layers.

A typical CNN architecture consists of a number of convolutional and subsampling layers followed by several fully connected layers. The convolutional layer contains several feature maps. Each unit in each of the feature maps is connected to a local subset of units in the feature maps of the previous layer. Mathematically speaking, each feature map is obtained by convolving the input with a linear filter, adding a bias, and then passing through a non-linear function. The subsampling layer usually computes the maximal value of a local subset of units in each feature map in the convolutional layer. This process not only reduces the computational complexity for subsequent layers, but also provides a certain degree of shift-invariance. The fully connected layers are traditional multilayer perceptron (MLP). The parameters of CNNs (weights and biases) are trained by using the backpropagation algorithm.

For our application, we used the open-source implementation named *cuda-convnet* which uses Graphical Processing Units (GPUs) to accelerate the computation speed. It was developed by Krizhevsky et al. [26, 27]. *cuda-convnet* provides options of various types of layers, hidden unit nonlinearities, etc. For example, in *cuda-convnet*, the schemes of local normalization and overlapping pooling can be used in a layer to improve generalization. In addition, a regularization method called *dropout* [28], whose key idea is to randomly drop units from the neural network during training, can be employed to reduce overfitting in the fully connected MLP layers. For the details of the architecture and the training protocol, refer to [26, 27].

3.2 Results on Open-i dataset

Among the 30,390 face regions extracted by the VJ detector, there were 9,357 true positives and 21,033 false positives. These face regions were manually labeled by one engineer by visual examination. We converted all the one channel grayscale images to three-channel images, and then resized all the images to 32 x 32. Figure 8 shows the architecture of the CNN we applied to our dataset. It contains two convolutional layers (*conv1* and *conv2*), two pooling layers (*pool1* and *pool2*), two locally-connected layers with unshared weights (*local1* and *local2*), a fully-connected layer (*fc*), and a soft max layer (*softmax*). For all the layers except the *fc* layer and the *softmax* layer, we employed rectified linear units (ReLUs) as the nonlinear function. For both convolutional layers, 64 filters of size 5 × 5 were applied; the distance between successive filter applications was set to be 1 pixel; the biases of every filter in the layer were set to be the same amongst all applications of that filter; the biases were initialized to be 0.5; the images were padded with a border of zeros of 2 pixels. Other parameters, such as the weights, were initialized from a normal distribution with mean zeros and standard deviation of 0.0001 for *conv1* and mean zeros and standard deviation of 0.01 for *conv2*; the parameter *partialSum* which affects the performance of the weight gradient computation was set to be 4 for *conv1* and 8 for *conv2*. Both pooling layers (*pool1* and *pool2*) are max-pooling layers; the size of the pooling region in either the x or y dimension was defined as 3; and the stride size between successive pooling squares was set to be 2 which means the overlapping pooling scheme was used. For both the locally-connected layers (*local1* and *local2*) which are convolutional layers but with no weight-sharing, 32 filters of size 3 × 3 were applied, and the standard deviation for the normal distribution used for initializing the weights was set as 0.04. The fully-connected layer (*fc*) has 2 outputs and other parameters for this layer were set with the default values. The final layer, a *softmax* layer, outputs a probability for each class. The learning parameters were set as follows: the weight learning rate, bias learning rate, weight momentum, and bias momentum for convolutional layers, locally-connected layers and the fully-connected layer were 0.0001, 0.002, 0.9,

and 0.9, respectively; the L2 weight decay value was 0 for convolutional layers and 0.004 for locally-connected layers and the fully-connected layer. The network used multinomial logistic regression as the object function to optimize. The face region data (30,390 regions) was divided into 10 batches, with each of the first 9 batches containing 1000 true positives and 2,200 false positives, and the last batch containing the reminder of the true positives and false positives. The first 8 batches (batch 0-7) were used for training and the last 2 batches (batch 8-9) were used for testing. Therefore, the training set contains 8,000 face regions and 17,600 non-face regions. The test set contains 1,357 face regions and 3,433 non-face regions. The number of epochs (one pass through the training data) for CNN training was 60. Figure 8 shows the classification error rate on the training set and the test set with the increase of the epoch number. The classification error rate for the test set is around 0.04 (i.e., classification accuracy is 96%) after 30 epochs. Table III lists the confusion matrix of the test set at the epoch 60. Figure 9 shows the trained filters of the first convolutional layer (*local1*), in which there exist edge patterns. Figure 10 shows the classification results of eight images randomly selected from the test set. The true labels of those images are shown in red. All of them were correctly classified except the last image, a non-face region, was misclassified as a face region.

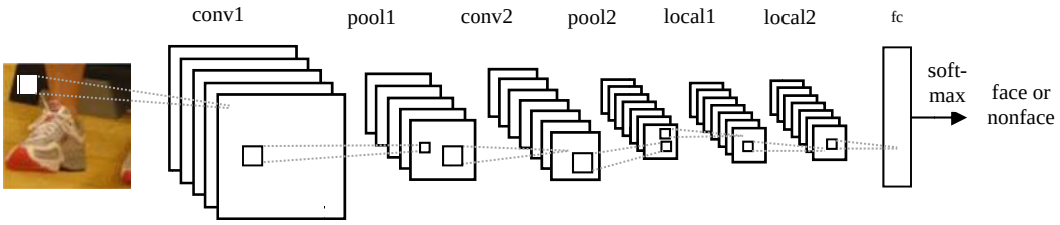

Figure 7. CNN architecture used for Open-i dataset

Table III. Classification results of CNN

		Predicted Class	
		Face	Non-Face
Actual Class	Face	1232	125
	Non-Face	62	3371

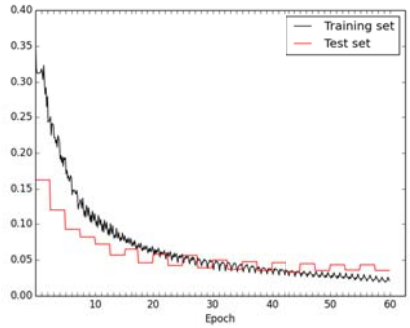

Figure 8. Classification error rate

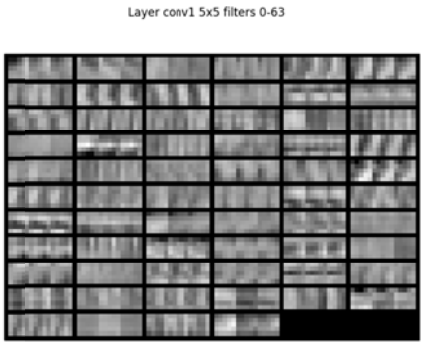

Figure 9. The trained filters of the first convolutional layer

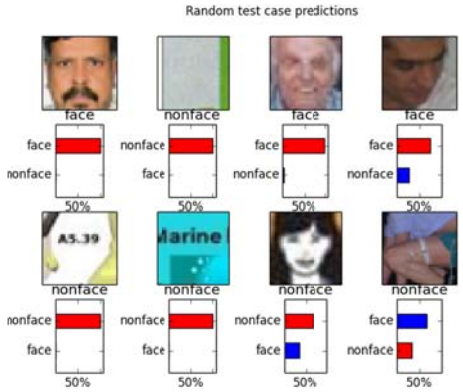

Figure 10. Classification predications of random test images

3.3 Comparison with the re-trained Viola-Jones detector

We used the same face region data that CNN used to train and test the VJ detector. For the training of the VJ detector, there are several important parameters: the number of cascade stages; the object size for training; the false alarm rate; the true positive rate; and the feature type. We set the number of cascade stages as 15, the acceptable false alarm rate at each stage as 0.2, the minimum true positive rate required at each stage as 0.995, and the feature type as HOG (histogram of oriented gradient). The training object size was determined automatically based on the median width-to-height ratio of the positive instances. For the description and trade-off of these parameters, please refer to the Mathworks website [29]. For training, all the face regions in the batch 0-7 were used as positive samples. Since the Viola-Jones object detector provided by Matlab takes negative images as input (it automatically generates negative training samples from the negative images by using sliding windows), we used the images from which the non-face regions in the batch 0-7 were extracted but do not contain any faces as negative images. For testing, all of the regions in the batch 8 and 9 were used as input images. Table IV lists the confusion matrix of the testing results. Comparing to Table III, the number of true positives is much lower for the VJ detector while the number of true negatives is almost the same. As a result, the classification accuracy of the VJ detector is 84.5% which is lower than that of the CNN (96%).

Table IV. Classification results of the re-trained Viola-Jones detector

		Predicted Class	
		Face	Non-Face
Actual Class	Face	673	684
	Non-Face	61	3372

4. FACE IMAGE EXTRACTION

Figure 11 shows the diagram of the whole procedure. Given an input photographic figure image, the first step is to use the pre-trained Viola-Jones face detector to extract face region candidates. The second step is to use the CNN to classify whether each of the candidates is a face region or not. If there is at least one region in the image classified by CNN as a face region, the input image is identified as a face image.

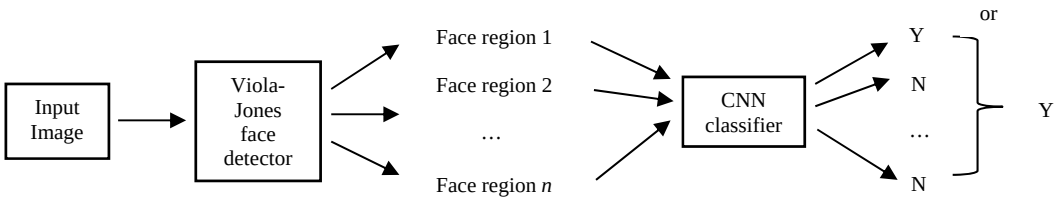

Figure 11. System diagram

5. CONCLUSIONS

In this paper, we present our method for extracting face images (image containing faces) from the figures in the photograph category included in the Open-i database. Due to the large number of images, it would be very time consuming to generate a ground truth dataset of images with face regions being marked. Instead, we applied a Viola-Jones face detector likely trained using the FERET database (but see Section 2.1) to our dataset and extracted candidate face regions. The candidate face regions obtained by this Viola-Jones detector contained many false positives. We then used the labeled candidate face regions to train and test a convolutional neural network, a classification method that does not require hand-engineered features but just the raw pixel values as input. The convolutional neural network achieved

96% classification accuracy for the test set, a significant increase in the precision of the detection. We also used the labeled candidate face regions to train a new Viola-Jones detector and compared its performance with that of the convolutional neural network. Therefore, for any given photographic figure, the system decides if it is a face image or not by applying the pre-trained Viola-Jones face detector first followed by the CNN classifier. By this approach, we avoided the intense labor work for manual delineation of face regions in a large dataset but substantially reduced the false positive rate of the system. Future work includes using the trained CNN to scan across the input image at different scales to identify face regions and comparing the performance of that method with the one presented in this paper.

ACKNOWLEDGEMENT

This research was supported by the Intramural Research Program of the National Institutes of Health (NIH), National Library of Medicine (NLM), and Lister Hill National Center for Biomedical Communications (LHNCBC).

REFERENCES

[1] You, D., Antani, S.K., Demner-Fushman, D., and Thoma, G. R., "A MRF model for biomedical image segmentation," Proceedings of the IEEE 27th International Symposium on Computer-Based Medical Systems, (2014).

[2] Xue, Z., You, D., Chachra, S., Antani, S.K., Long, L.R., Demner-Fushman, D., and Thoma, G. R., "Extraction of endoscopic images for biomedical figure classification," Proc. SPIE. 9418, Medical Imaging 2015: PACS and Imaging Informatics: Next Generation and Innovations, 94180P (March 2015)

[3] Zhang, C., Zhang, Z., "A survey of recent advances in face detection," Technical Report, MSR-TR-2010-66, Microsoft Research, (2010).

[4] Viola, P., Jones, M.J., "Robust real-time face detection," International Journal of Computer Vision, 57(2), 137-154 (2004).

[5] Jain, V., Learned-Miller, E., "FDDB: a benchmark for face detection in unconstrained settings," Technical Report, UM-CS-2010-009, Dept. of Computer Science, University of Massachusetts, Amherst. (2010).

[6] Rowley, H., Baluja, S., Kanade, T., "Neural network-based face detection", IEEE pattern Analysis and Machine Intelligence, 20, 22-38 (1998).

[7] Milborrow, S., Morkel, J., Nicolls, F., "The MUCT landmarked face database," Pattern Recognition Association of South Africa, (2010).

[8] LeCun, Y., Bengio, Y., Hinton, G., "Deep learning", Nature, 521, 436-444 (2015).

[9] Arel, I., Rose, D.C., Karnowski, T.P., "Deep machine learning – a new frontier in artificial intelligence research," IEEE Computational Intelligence Magazine, 14 – 18 (November 2010).

[10] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., "Gradient-based learning applied to document recognition," Proceedings of the IEEE, 86 (11), 2278–2324 (1998).

[11] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision, (2015).

[12] Freund, Y., Schapire, R., "A decision-theoretic generalization of on-line learning and an application to boosting," Journal of Computer and system Sciences, 55(1), 119-139 (1997).

[13] Mita, T., Kaneko, T., Hori, O., "Joint Haar-like features for face detection," Proceedings of the International Conference of Computer Vision, 2, 1619 – 1626 (2005).

[14] Meynet, J., Popovici, V., Thiran, J.P., "Face detection with boosted Gaussian features," Pattern Recognition, 40(8), 2283-2291 (2007).

[15] Jin, H., Liu, Q., Lu, H., Tong, X., "Face detection using improved LBP under Bayesian framework," Proceedings of the 3rd International Conference on Image and Graphics, 306-309 (2004).

[16] Wang, X., Han, T.X., Yan, S., "An HOG-LBP human detector with partial occlusion handling," Proceedings of International Conference of Computer Vision, (2009).

[17] Lienhart, R., Kuranov, A., Pisarevsky, V., "Empirical analysis of detection cascades of boosted classifiers for rapid object detection," Proceedings of the 25th DAGM Symposium on Pattern Recognition. Magdeburg, Germany, (2003).

[18] Heisele, B., Serre, T., Prentice, S., Poggio, T., “Hierarchical classification and feature reduction for fast face detection with support vector machines,” Pattern Recognition, 36, 2007-2017 (2003).
[19] Garcia, C., Delakis, M., “Convolutional face finder: A neural network architecture for fast and robust face detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(11), 1408-1423 (2004).
[20] http://www.mathworks.com/help/vision/ref/vision.cascadeobjectdetector-class.html
[21] Freund, Y., Schapire, R.E., “Experiments with a new boosting algorithm,” Proceedings of the Thirteenth International Conference on Machine Learning, 148-156, (1996).
[22] http://note.sonots.com/SciSoftware/haartraining.html
[23] http://www.itl.nist.gov/iad/humanid/feret/feret_master.html
[24] Berg, T. L., Berg, A.C., Edwards, J., Forsyth, D. A., “Who’s in the picture,” Neural Information Processing Systems (NIPS), (2004)
[25] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D., “Handwritten digit recognition with a back-propagation network,” Neural Information Processing Systems (NIPS), 296-404, (1990).
[26] Krizhevsky, A., Sutskever, I., Hinton, G., “ImageNet classification with deep convolutional neural networks,” Neural Information Processing Systems (NIPS), 1097-1105 (2012).
[27] https://code.google.com/p/cuda-convnet/
[28] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. “Dropout: a simple way to prevent neural networks from overfitting,” The Journal of Machine Learning Research, 15(1), 1929-1958 (2014).
[29] http://www.mathworks.com/help/vision/ref/traincascadeobjectdetector.html