CNNs for Face Detection and Recognition

Yicheng An
Department of Electrical Engineering
Stanford University
yicheng@stanford.edu

Jiafu Wu
Department of Electrical Engineering
Stanford University
jiafuwu@stanford.edu

Chang Yue
Department of Electrical Engineering
Stanford University
changyue@stanford.edu

Abstract

Currently face detection method is becoming a more and more important technique in our social lives. From face detection technology implemented in our cheap cameras to intelligent agencies' sophisticated global skynet surveillance system, such techniques have been widely used in a large number of areas and the market is still growing with a high speed. Face detection has been an active research area with many successful traditional and deep learning methods. In our project we are introducing new convolutional neural network method to tackle the face detection to maintain a high accuracy while running in real time.

1. Introduction

1.1. Problem Statement

In our imaginary scenario, our technique could be used by companies and facilities with security levels: customers could put all personnel's info into the dataset, and then put the camera at the desired position like the office front gate. If people coming in are not in the dataset, there would be an alarm; otherwise, everything should be fine.
Therefore, our method has mainly 2 steps and 1 requirement, the first step should be localization, which means the system could localize faces and then circle them out in photos or videos. The second step could be the classification process, once faces get circled out, we need to tell whether this person belongs to our dataset, if so, who he/she is. If not, just classify him/her as unknown and sound the alarm.
Then our largest requirement would be to complete both localization and classification processes in real time, otherwise it would be no use as a security camera package.

1.2. Plan

Based on the steps and requirement mentioned above, we split the whole project into several sections. The first part would be finding the regions that would potentially contain faces. There are a lot of methods doing this job but we need to come up with a new one by ourselves to maintain the real time processing requirement.
And the second section would be classifying these obtained potential regions to get identification numbers of people in these regions. Then the third section would be testing its detection and classification accuracies and its processing time in order to make sure our algorithm could do the task in real time. The final section would be tuning the parameters to make the accuracy even higher, which was a pretty time consuming procedure.

2. Related Works

Face detection has been an active research area since the development of computer vision, and many classical and deep learning approaches have been applied in this field. Particularly, similar to many other fields in computer vision, deep learning approach using neural network has achieved significant success in tackling face detection as a subclass of object classification, localization, and detection. Apparently, the evolve of face detection correlates closely with the development of object classification, localization and detection techniques.

2.1. Sliding Window

In the early development of face detection, researchers tended to treat it as a repetitive task of object classification, by imposing sliding windows and performing object classification with the neural networks on the window region. Vaillant et al.[6] had proposed a method to train a

convolutional neural network to detect the presence or absence of a human inside an image sliding window area and scan the whole image with the neural network on the sliding window region for all possible locations. This method can be easily tackled by today's state-of-the-art techniques but it can be viewed as an early approach of utilizing the sliding window technique to detect the human face. Then face detection techniques gradually evolved to extend for rotation invariant face detection with a network to estimate the face orientation in order to apply the proper detector network with the corresponding face orientation [7]. Then the trend got shifted to Convolutional Neural Network after CNNs have achieved significant breakthrough on image classification and object detection [8], and the mainstream face detection methods have all turned to CNN-based object detection algorithms. Nevertheless, the sliding window approach still needs to apply CNN on many different sliding windows and it is still a repetition of performing image classification on local regions; as a result, it is extremely computationally expensive with repetitive computations of CNNs.

2.2. Detection with Proposals

In order to handle the expensive computation problem, instead of performing CNN computation many times on every sliding window location, people tried to find a way to reduce the candidate locations of the sliding window. As a result, region proposal method was developed to find potential regions that have a high possibility containing objects[10], through which the number of potential regions is reduced compared with the sliding window approach.

One of the most significant breakthrough on object detection with Region Proposals is the R-CNN developed by Girshick et al. [9]. First R-CNN generates approximately 2000 Regions of Interest (RoI) using the Region Proposal method on the input image, then it warps each RoI into standard input size for the neural network and forward them into the CNNs dedicated for image classification and localization and output the class category as well as the bounding box coordinates and sizes. However, R-CNN method still has many problems even after it used the region proposals. For example, the training time is slow, which takes 84 hours on the PASCAL VOC datasets and it consumes many memory space [9]. Moreover, during the testing, the detection is also slow; in average, it takes 47 seconds per image with VGG16 model [11].

More advancements based on R-CNN network occurred to deal with the expensive slow run time problem, such as Fast R-CNN [12] and Faster R-CNN [13]. Fast R-CNN forward the whole image through the CNN at the beginning so that it is only performed once instead of many times in R-CNN [12]; Faster R-CNN performed RoI pooling and make the CNN to do the Region Proposal, which inserts the Region Proposal Network (RPN) as part of the layers in the CNN model to predict the possibility of objectiveness in the region [13].

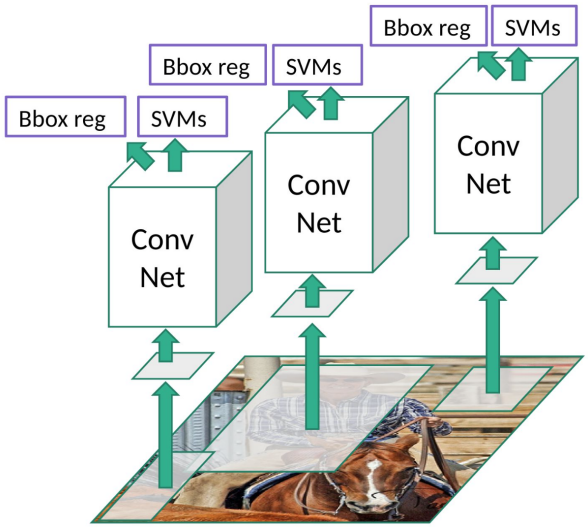

Figure 1. The architecture of the R-CNNs with Region Proposals [9]

While they achieve significant reduce in training time and test time compared with the R-CNN methods [12] [13], the runtime is still dominated by region proposals method.

2.3. Detection without Proposals

Some detection methods were introduced without the Region Proposal method in order to achieve smaller training and testing time. YOLO [1] and SSD [5] are two detection methods without Region Proposals. The input image goes directly to one big convolutional neural network. Inside the network, the input image at first is divided into many grid cells, and the classification scores and the bounding box coordinates and scales are determined on each grid cell. And the overall object classes and bounding boxes are calculated based on the results obtained from each grid cell. These two approaches further reduce the training and test time time but the accuracy is compromised compared with the method using Region Proposals [14].

2.4. Comparison

Overall, based on the previous related work, we found out that the sliding windows technique has the lowest difficulty in implementation since it is essentially a repetition of performing image classification task but it would be extremely slow during the training and testing time, while the accuracy relies on the maturity of the network that performs the image classification. The region proposal method has a reduce on the training and testing time compared with the sliding window techniques and helps increase the detection

accuracy; in fact, faster R-CNN achieves the highest accuracy compared with other methods [14]. In terms of training and testing time, SSD is significantly faster than other methods since it gets rid of the Region Proposal method, but with a cost of reduced accuracy compared with those with Region Proposal.

3. Methods

For our project, we developed our face detection methods using the following approaches:
First, we developed a model called Two Stream CNN, which specializes in classification and localization for a single face detection. With an input image, this Two Stream CNN method is capable to output whether it contains a human face or not, and if there is a human face, it would also output the identity of that human (classification) as well as the coordinate and the size of the bounding box (localization).
We also try to perform our multi-object detection with a cascade of Region of Interest Network and Two Stream CNN. The Region of Interest Network helps reduce the number of repetitions.

3.1. Two Stream CNN

For our face detection problem, we first tried to simplify it into a simpler problem as a single face detection problem. We would construct our network that is capable to detect a single human face and output the coordinate and size of the bounding box as well as the class (human identification or no human). That is, we first simplify the face detection problem as a classification and localization problem.
Our model utilizes CNN to do classification and localization in a single evaluation. Our model consists of 6 primary modules; each module has one convolution layer, one max pooling layer and one leaky ReLU layer except the very last module. Besides, we also added a few dropout layers between modules for regularization. In order to make prediction, we feed the result of the last module into two sets of fully connected layers, one for predicting the location and the size (center, width, height) of human faces, another one for predicting the identity of the person. Our network is intelligent enough to tell whether there are people in the scene, and if true, who the person is. Since our objective of this Two Stream CNN is to classify and localize single human face, in case of multiple people showing up, the network selects the nearest one to the camera.

3.2. Cascade CNN

While our Two Stream CNN dedicates to perform single face detection, it is essentially a classification and localization on single face only and is unable to tackle the image with multiple faces. As a result, inspired by the region proposal method and sliding window method, we would duplicate this single face detection algorithm cross candidate location of the image.

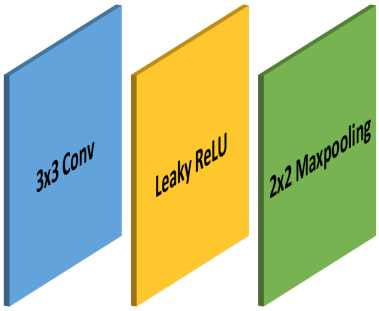

Figure 2. The basic architecture of each module

First, we perform sliding window across the whole image and each sliding window is 48 pixels by 48 pixels, as shown on Figure 4. We choose this number due to the sizes of the human faces in our dataset; this window with such size is capable to cover most human faces in our dataset. The window is slided across the whole image with stride 24. The content of each window is fed into a convolutional layer for binary classification which can detect whether there is human face or not on the image and in this stage, we only care about whether there is a human face exist inside our window. We set a threshold on the output score of the layer and if the score of the sliding window content exceeds the threshold, we set that sliding window as our candidate region for our next stage. In our second stage, non-maximum suppression(NMS) is used to eliminate highly overlapped detection region, in order to reduce the repetition of operations. We perform another sliding window across the candidate region from our first stage. In this stage, the window size is doubled into 96 pixels by 96 pixels, while the stride is 48 pixels. Again, similar to the first stage, we perform the binary classification with a convolutional layer on the sliding window region. We set a threshold on the score and set the region with scores above threshold as our candidate region for our next stage.
In our final stage, we perform non-maximum suppression again to remove overlapped regions. Then on those final regions, we performed our Two-Stream CNN to obtain the class as well as the more detailed bounding box coordinates and size, which will be our final output of classes and bounding boxes for our input image.
This cascade technique is used to help us tackle multi-object detection problem with our single face classification and localization method, while the Non-Maximum Suppression significantly helps reduce the repetition of operation compared with simply sliding across all regions. Overall, it helps us to tackle the multi-object detection while maintains a relatively high computational time by narrowing down number of candidates during each step.

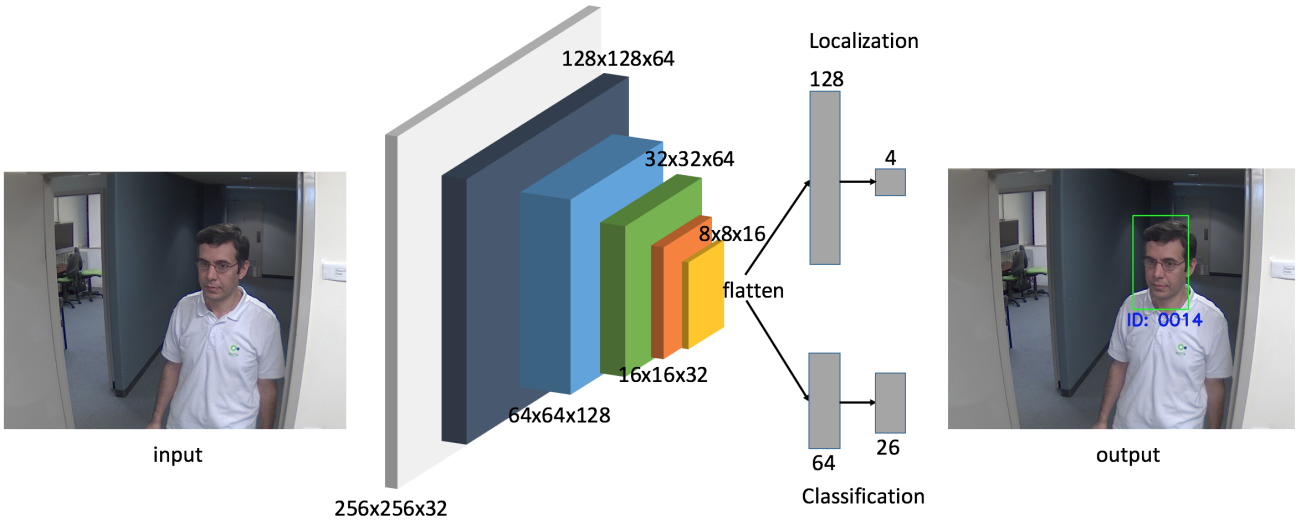

Figure 3. The simplified architecture and framework of Two Stream CNN

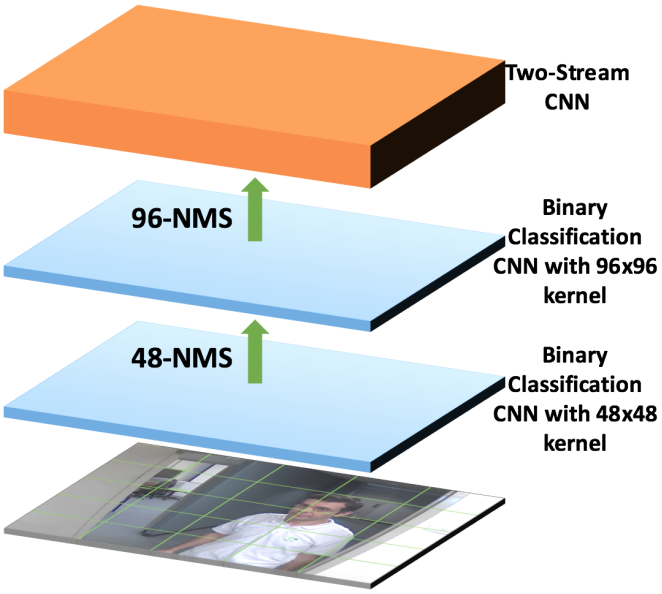

Figure 4. The framework of Cascade CNN

3.3. Loss Function

Our loss function consists of two parts, the coordinate loss and the size loss of the bounding box as well as the classification loss. The coordinate loss and the size loss measure the squared L2-norm of the differences between the prediction and the groundtruth of (x, y), w, h, while the classification loss is the Softmax cross-entropy loss of the classification over 26 classes. Another important idea is to assign different penalties for different types of losses and ignore center coordinate loss if no person show up. The formula of the loss function is shown below:

$$\frac{\lambda_{corrd}}{N}\sum_{i=0}^{N}\mathbb{1}_i^{obj}\left[(x_i-\hat{x_i})^2+(y_i-\hat{y_i})^2\right]$$

$$+\frac{\lambda_{size}}{N}\sum_{i=0}^{N}\left[(w_i-\hat{w_i})^2+(h_i-\hat{h_i})^2\right]$$

$$-\frac{\lambda_{class}}{N}\sum_{i=1}^{N}\log\left(\frac{e^{f_{c_i}}}{\sum_j e^{c_j}}\right)$$

3.4. Training

We choose a minibatch size of 32 frames of images due to the limitation of our GPUs. We picked Adam optimizer and performed our training for approximately 2500 iterations. The loss decay is shown in Figure 5.

3.4.1 Transfer Learning

In our project we were using VGG-16 to pretrain our model first, in the training process the loss decreased pretty fast which was fine but the predicting result was not significant advanced, probably because the network was particularly trained on ImageNet dataset and not specialized in classifying different human identities. Therefore, we chose to train from scratch, which could make the network far simpler.

4. Experiments

4.1. Dataset

In order to train a convolutional neural network, we need a relatively large dataset. Here the word ”large” has two

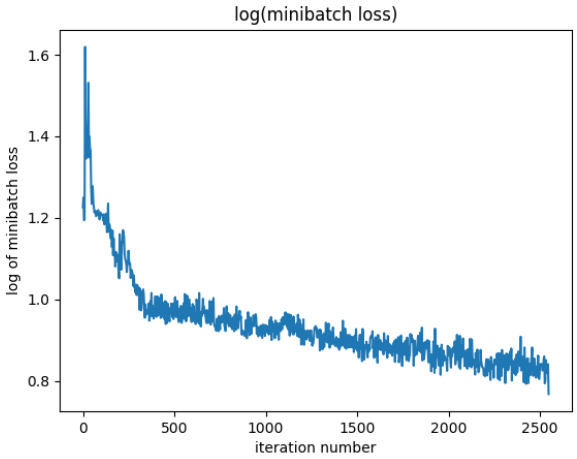

Figure 5. The training loss over number of iterations

Figure 6. Three examples of the dataset

meanings, the first meaning is that the dataset has to contain enough images to prevent potential overfit. The second meaning is that there should be multiple people within the dataset, since our face detection is specialized to detect multiple human face in the scenes, otherwise the classification accuracy would be highly inaccurate.
In our project the dataset we were using was called ChokePoint[17], which contained frames of one camera placed above an entrance in indoor environment, with scenes containing different people walking by in natural pose.
The dataset consists of 25 subjects (19 male and 6 female), having a frame rate of 30 fps, and the image resolution is 800X600 pixels. In total, the dataset consists of 48 video sequences and 64,204 face images, which is a reasonably large number to train our CNN.
Still one drawback of this dataset is that it only provides ground truth labels of centers of eyes, instead of center points, widths and heights of bounding boxes. We have to compute the ground truth boxes we need from a human body formula [15] [16]. And the other drawback is that many of the images in the dataset are empty, with no people in it, which could have some effect on the training result.

4.2. Setup

We develop our algorithms on Python (our code is compatible to both Python2 and Python3). And we employ Tensorflow 1.1.0 to build the network, loss function and solver, which saves us a lot of effort. We also imported cv2 for loading and writing images, drawing boxes and other small image processing methods. We trained our models on NVIDIA Tesla K80 GPU for 2500 iterations, actually the GPU saved us a lot of time during the training process.

4.3. Evaluations

We consider all indoor scenarios: nobody shows up; somebody appears but none belongs to the group; one person from the group shows; more than one person show up. For the first two cases, we classify it as 'others'. And for the last two cases, we count everyone belonging to the group.
For a security camera, the most important feature is to correctly identify whether or not the person belongs to the group, so we will measure the false positive and false negative rates; the false positive is the probability that the people outside of the dataset get classified as the people inside the dataset, while the false negative is the probability that people inside the dataset get classified as the people outside of the dataset. Since we also want our camera to tell the ID number of the person, we introduced another metric named Accuracy to measure the classification accuracy. We also have 2 more metrics to measure the bounding box location accuracies, named average IoU (intersection over union) and box center deviation.
Computation time is also crucial, for cameras real-time nature. We use FPS (frames per second) as the metric. There is a trade-off between classification accuracy and bounding box accuracy, and this could be tuned in the loss function.

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\{f_{ci} == c_i\}$$

$$\text{Average center deviation} = \frac{1}{N}\sum_{i=1}^{N}||x_i - \hat{x_i}, y_i - \hat{y_i}||_2$$

$$\text{Average IoU} =$$

$$\frac{1}{N}\sum_{i=1}^{N}\frac{\text{Intersection of prediction box and ground truth}}{\text{Union of prediction box and ground truth}}$$

4.4. Results

4.4.1 Statistics

The results of our measurement are shown below:
False Positive = 0.0884%
False Negative = 3.37%
Accuracy = 91.0%
Average IOU = 58.7%
Average center deviation = 11.3 pixels
Frames per Second = 11 on CPU, 89 on GPU
Our false positive is extremely small, which matches with the purpose of our security camera to block the outsider to get access into our facility; nevertheless, the trade-off

of such low false positive is that a relatively higher false negative, which means sometimes our insiders might get blocked to get access. The overall accuracy, which measures the correctness of our classification network, is also pretty high which means in most cases our system can get each individual classified with his/her identity. The average IOU is 58.7137%, which is also a pretty decent result to get the bounding boxes overlapped with the ground truth.

4.4.2 Visualization

Figure 7 shows the result of the Two Stream CNN, which is the classification and localization result for a single human face detection.
Figure 8 shows one good example of the step-by-step outputs of the Cascade CNN. Firstly, after the binary classification layer with 48 by 48 kernel, many potential regions of human faces were generated and they are highly overlapped. After the Non-Maximum Suppression(NMS), the number of potential regions were reduced. In order to further reduce the number of potential regions, we perform another binary classification with 96 by 96 kernels followed by Non-Maximum Suppression to reduce duplicated candidate regions. After that, we would perform the Two Stream CNN on the final candidate regions to obtain the identities of human and more accurate bounding boxes.
While it seems like the number of potential regions, shown in the figures as green boxes, did not reduce during the 96x96 classification process, and the output did not change after the 96-NMS step compared with its input, it is because this example displays an optimal case that the number of potential regions has been reduced by the 48 by 48 kernel Non-Maximum Suppression. In fact, lots of redundant detections are eliminated during the 48-NMS process. The second stage of 96 by 96 binary classification and Non-Maximum Suppression is still needed in case that the 48 by 48 kernel is unable to suppress the duplicated candidate regions.

5. Conclusion

5.1. Summary

In this project we have done a lot of research on related algorithms for face detection such as LSTM, R-CNN and YOLO before we actually started to implement our own version of neural network, and then we chose to extract some good aspects of these well-developed algorithms and made our own innovations. Since all these mentioned methods have their own strengths and drawbacks, we would like to combine them to achieve an optimal performance to achieve our goal for higher accuracy and lower run time.
And through the process of doing the project, we have managed to overcome numbers of problems, like how to generate potential regions more effectively, how to compensate the locations and the size of bounding boxes more accurately.

Figure 7. The output examples of the Two Stream CNN

5.2. Future

After the course we will explore more sophisticated proposal region generation methods to further increase the detection speed. And then we will try to combine the good aspects of faster RCNN and other convolutional neural network architectures in order to increase our detection and classification accuracies.
Also, during the poster session many visitors asked us about what if you have new people join in the dataset? Do you have to re-train everything again? We found this question extremely valuable and we are having some idea to solve it now (e.g., adding residual networks), in the future we would like to fully explore solutions and try to provide a good answer to this question.

5.3. Statement

5.3.1 Open Source

Here is our GitHub link for our project: `https://github.com/fusio-wu/CS231N-Final-Project`

5.3.2 Contribution

We have 2 team members — Yicheng An and Chang Yue who are also taking CS231A this quarter, we are in one team with another guy, Chao Wang, doing self driving project using a convolutional neural network built by ourselves, which is a completely different approach from what we were doing in this face detection project. The GitHub

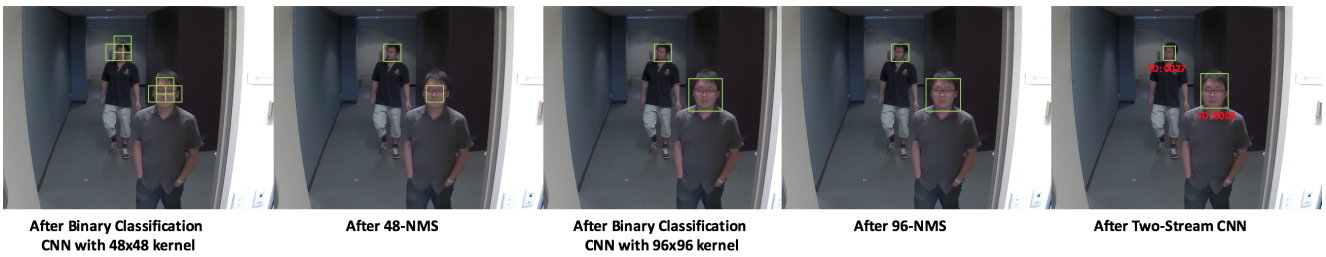

Figure 8. The output of each step of the Cascade CNN

link for that project is here: `https://github.com/Flyhigh2017/CS231A_project`

References

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi(2016). You Only Look Once: Unified, Real-Time Object Detection. 2016 CVPR

[2] Joseph Redmon, Ali Farhadi(2017). YOLO9000: Better, Faster, Stronger. 2017 CVPR

[3] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi(2016). XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. 2016 ECCV

[4] Y. Wong, S. Chen, S. Mau, C. Sanderson, and B.C. Lovell. Patch-based Probabilistic Image Quality Assessment for Face Selection and Improved Video-based Face Recognition. 2011 CVPR

[5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single Shot MultiBox Detector. 2016 ECCV

[6] R. Vaillant, C. Monrocq, and Y. Le Cun. Original approachfor the localisation of objects in images.IEE Proceedings-Vision, Image and Signal Processing, 1994.

[7] H. A. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. InComputer Visionand Pattern Recognition, 1998.

[8] Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh,S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein,A. C. Berg, and L. Fei-Fei. ImageNet Large Scale VisualRecognition Challenge, 2014.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. 2013.

[10] Alexe et al, Measuring the objectness of image windows, TPAMI 2012.

[11] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556.

[12] Girshick. Fast R-CNN. InICCV, 2015.

[13] Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towardsreal-time object detection with region proposal networks. InNIPS,2015.

[14] Huang et al, Speed/accuracy trade-offs for modern convolutional object detectors, CVPR 2017.

[15] Gordon, C. C., Blackwell, C. L., Bradtmiller, B., Parham, J. L., Barrientos, P., Paquette, S. P., Corner, B. D., Carosn, J. M., Venezia, J. C., Rockwell, B. M., Murcher, M., Kristensen, S. (2014).2012 Anthropometric Survey of U.S. Army Personnel: Methods and Summary Statistics. Technical Report NATICK/15-007. Natick MA: U.S. Army Natick Soldier Research, Development and Engineering Center.

[16] Mester, Jessica L et al. Analysis of Prevalence and Degree of Macrocephaly in Patients with GermlinePTENMutations and of Brain Weight inPtenKnock-in Murine Model.European Journal of Human Genetics19.7 (2011): 763768.PMC. Web. 30 May 2017.

[17] Y. Wong, S. Chen, S. Mau, C. Sanderson, B.C. Lovell Patch-based Probabilistic Image Quality Assessment for Face Selection and Improved Video-based Face Recognition IEEE Biometrics Workshop, Computer Vision and Pattern Recognition (CVPR) Workshops, pages 81-88. IEEE, June 2011.