Ear detection with Viola-Jones

Assignment #2

Image Based Biometrics 2018/19, Faculty of Computer and Information Science, University of Ljubljana

Robert Cvitkovič

***Abstract*—We compare RefineNet with Viola-Jones for ear detection. We show how to use and train Viola-Jones models with the help of the open-source OpenCV library. During our testing we discover that the intersection over union for Viola-Jones is only 0.22 but also note that this result is gained from training on only a few hundred images.**

I. Introduction

A biometric pipeline consists of four stages: data acquisition, feature extraction, matching and labeling. An impotent part in the data acquisition is the separation of the desired biometric modality from the background. The rest of the pipeline works best if the input is just the subsection where the modality is located.

We will focus on ears for whiche there are many different approaches for detection (see [1] and [2]). We will compare the result from the convolution neural network RefineNet [3] with those from a more classical aprouche called Viola-Jones detector [4] which is besed on Haar-like featurs. Additional we will show how to use an open-source implementation of Viola-Jones from the OpneCV [5] library and compare a prelearned models with our own models.

II. Methodology

A. RefineNet

The results from the RefineNet detection were given to us from the assistant. To measure the correctness of the detection we first needed the grand truth. This was achieved by manually reviewing the detection and annotating the images in which the ears were correctly detected. To speedup the process we created a simple graphical user interface (gui) program that displayed the cropped image of the ear and two buttons to annotate the image with either correctly detected or not.

An additional assignment was to annotated with right or left for which ear was on the images. This was again done manual but with the help of another simple gui program.

B. Viola-Jones

We accomplished our detection with the help of the OpneCV library. It is a widely used open-source image and video processing library which supports the most often used programing languages like Python, C++, Java and more. The Viola-Jones algorithm is implemented in the class *CascadeClassifier* for which the initial input is the location of the cascade classifier model. The class implements the method *detectMultiScale* whose input are image and algorithm parameters.

1) OpenCV models: OpenCV contains many prelearned models for detecting different kinds of objects like full body, faces, eyes, ears and more. We used two models for left and right ear. On each image we run both classifiers and return the area with the highest certainty of classification.

2) Custom models: The OpenCV library also contains commend-line interface (cli) tools to build custom cascade classifiers. To train a model we need atleast one picture of the desired object but for the best result more is better. Additionally we need images of anything except for the desired object. To build the classifier file we first used the cli tool *opencv_createsamples* to prepare sample images and then *opencv_traincascade* to train the model. For training we used two different set of parameters and thereby created and tested two custom set of models. One is trained on sample images that are equal in hight and width and the second on samples that are double in hight than in width. The second one represents the true size of ears more closely. For each set of parameters we trained models separate for left and right ear. The detection was done in the same way as with the prelearned models. We used our dataset for detection and another for learning. The learning dataset contained 336 images of left ears, 256 images of right ears and 1000 images without ears.

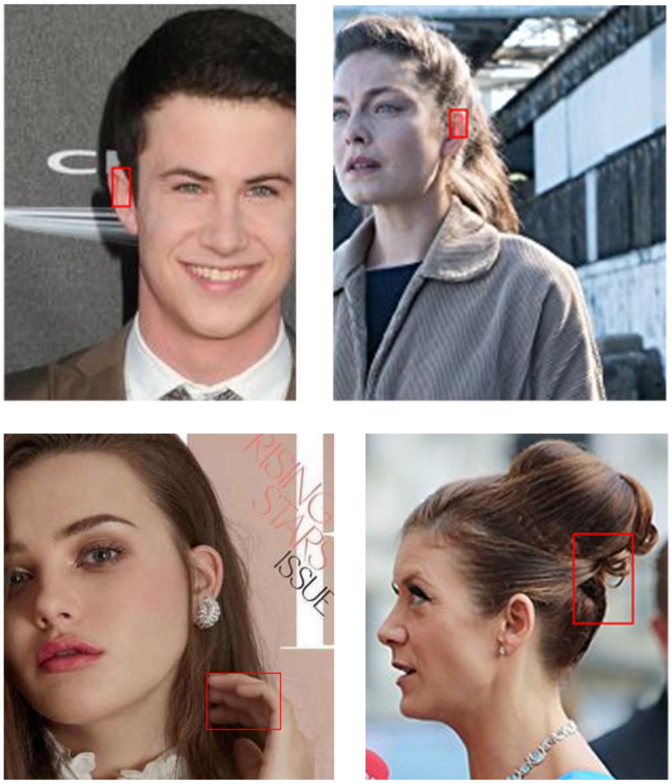

Figure 1: Samples of incorrect detections from RefineNet.

III. Results

A. RefineNet

RefineNet correctly detected 488 ears out of 600 images. On closer inspection of the 112 miss-detected ears we can observe two types of miss-detections: partial detection and complete miss-detection. On the two upper images in Figure 1 we can observe partial detection. That means that the ear was correctly located but the area didn't include the whole ear.

On the lower two images in Figure 1 we can observe complete miss-detection of ears. We can also observe that the detected regions contain mostly hair which is logical because ears are normally surrounded by hair. In the miss-detected regions we could also observe object that were half moon shaped, which is also logical considering that ears have the same shape.

B. Viola-Jones

To compare ReineNet to Viola-Jones we pretend that the correctly detected ears from RefineNet are the grand truth. This allows us to numerically evaluate the difference between detectors. We define intersection over union as:

$$IoU(A, B) = \frac{A \cap B}{A \cup B},$$

where A is the true area of the ear and B is the detected area. As Viola-Jones detects only bounding boxes (as seen on Figure 2) we used those as the area of the ear.

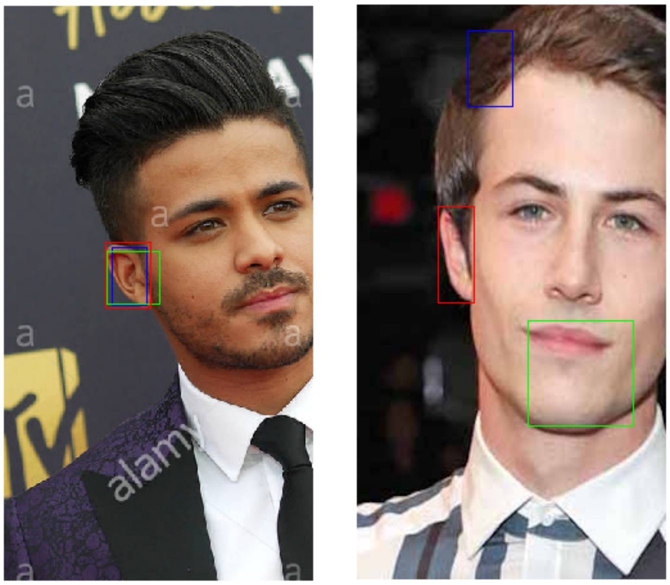

Figure 2: Ear bounding boxes as returned by the detectors: RefineNet - red, OpenCV model - purple and Custom model - green

To find the best detection we tried different values for parameters $scaleFactor$ and $minNeighbors$ which are set on $detectMultiScale$ call. The best set of parameters and the resulted IoU are displayed in Table I.

model	scaleFactor	minNeighbors	IoU
OpenCV	1.02	1	0.287
Custom 32x32	1.1	15	0.028
Custom 24x48	1.05	3	0.220

Table I: Results of testing different Viola-Jones models.

It is also important to look at the training time for the custom models. We measured it took 15 minutes for each ear to train on the equal sized samples and 3 hours for each ear on the unequal sized samples. The learning was done on six cores of an Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz and 40GB of RAM.

IV. Conclusion

We showed how to create a custom Viola-Jones based detector for a desired object. Although the result is not as good as that from RefineNet we achieved the results with just a few hundred annotated images. We purpose to improve the result by optimizing the training parameters and by using even more sample images.

References

[1] A. Pflug and C. Busch, “Ear biometrics: a survey of detection, feature extraction and recognition methods,” *IET biometrics*, vol. 1, no. 2, pp. 114–129, 2012.

[2] A. Abaza, A. Ross, C. Hebert, M. A. F. Harrison, and M. S. Nixon, “A survey on ear biometrics,” *ACM computing surveys (CSUR)*, vol. 45, no. 2, p. 22, 2013.

[3] G. Lin, A. Milan, C. Shen, and I. D. Reid, “Refinenet: Multi-path refinement networks for high-resolution semantic segmentation.” in *Cvpr*, vol. 1, no. 2, 2017, p. 5.

[4] P. Viola and M. J. Jones, “Robust real-time face detection,” *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[5] G. Bradski, “The OpenCV Library,” *Dr. Dobb's Journal of Software Tools*, 2000.