Improving Viola-Jones full-body detection

Assignment #2

Image Based Biometrics 2019/20, Faculty of Computer and Information Science, University of Ljubljana

Jan Pavlin

***Abstract*—People detection is an important aspect for biometrics, video surveillance and human computer interaction. In this assignment we try to replace Viola-Jones full body detector with combined face, upper body and lower body detectors.**

I. Introduction

In their paper Paul Viola and Michael Jones in 2001 [1] proposed a method of object detection using Haar feature-based cascade classifiers. The main property of this algorithm is that training is slow, but detection is fast.

All human faces share some similar properties. These regularities may be matched using Haar Features. Haar features are divided in rectangles and the value of each rectangle is calculated as sum of pixels in black area minus sum of pixels in white area. Viola & Jones uses two-rectangle features.

Example of two-rectangle Haar feature [2]

II. Methodology

Our goal was to use Haar cascades for detection of face, upper body and lower body. With that detection the model had to decide if there is full body on picture. We tried was to achieve similar or better result than full body detector. We decided not to use eye detector since eyes took very little proportion of image and they would be almost undetectable with a lot of false positive defections.

A. Dataset

To evaluate our model we used INRIA Person Dataset [3]. From this dataset we used images of people who are standing or walking. After we downloaded images we further inspected them and removed them if they were blurry. For final evaluation we used a set of 460 images.

B. Algorithm

Whole algorithm was implemented in python using OpenCV library. After the image was read, it was resized to fixed width of 600 pixels. Detectors then detected all faces, upper and lower bodies and marked them with different color. After we got all the detected features, it was time to make a decision if full body is present on image. We set a threshold that at least 2 features should be found in same vertical line to identify the full body. Also, the features had to be in logical order (face is smaller than upper and lower body, face is above lower body and within upper body, upper body is above lower body). After the algorithm decided that in found full body we draw a rectangle around it.

C. Detectors

For detectors of face, lower body and upper body we called function detectMultiScale, where the parameters were:

- in_image - Matrix containing an image where objects are detected
- scaleFactor - Parameter specifying how much the image size is reduced.
- minNeighbours - Parameter specifying how many neighbours each cadidate should have.
- minSize - Minimum possible object size

When all detections were finished, it was necessary to define what to classify as a full body. We decided, that detector should detect at least two different classes which should be logically located (head is within upper body and above lower body, upper body is above lower body) to be classified as a full body.

In code, there are three parameters that can be changed when running the algorithm:

- show_full - when True, image will have also white rectangles for detected faces, orange rectangles for detected upper bodies and blue rectangles for detected lower bodies.
- show_images - when True, the detector will show images one by one.
- save_results - when True, the results will be saved in folder 'results/'

III. Results

With our detector we managed to detect peoples relatively as good as full body detector. Detector became less efficient when faces of people were covered or turned away. Some of results can be seen on picture below where original detector shows green rectangle and our detector shows red rectangle:

We can see that in most cases our method was able to detect person just like full body detector.

Because even full body detector had many of false positive or false negative predictions, we decided to go through full data set by hand and generate confusion matrix as:

- True Positive - Detector detected a person that is on the image,
- False Positive - Detector detected a person that is not on the image,
- False Negative - Detector did not detect a person that was detected with full body detector and person was on the image,
- True Negative - Detector did not detected a person that was detected with full body detector and person was not in the image.

		prediction outcome		
		p	**n**	**total**
actual value	**p′**	198	178	376
	n′	17	93	110
	total	215	371	

Detector also detected 54 people that were on the picture but were not detected with full body detector. This number is already included in true positive.

IV. Conclusion

Full body detection using detectMultiScale function depends on size of the image and its shape. Since we had images of different sizes and shapes, we decided that all images should have same width of 600 pixels but the ratio between height and width will be preserved. If we would have all images of same size or we would evaluate model on only one image, we could tune in some parameters to get better results.

In confusion matrix we can see that number of times when detector found a person was almost half less than number of times when detector did not detect a person. This was mainly because in many pictures, it failed to detect face. The reason why our detector works worse if person has covered face or is turned around is because it has to correctly identify both upper and lower body, face is undetectable. Failure to identify face on blurry or dark images is one of reasons why there is so much false negatives. Since our model has 197 true positive and only 17 false positive predictions (which gives it 92% specificity) we can assume that whenever it detects a person on image, it is almost always correct.

References

[1] P. Viola, M. Jones *et al.*, "Rapid object detection using a boosted cascade of simple features," *CVPR (1)*, vol. 1, no. 511-518, p. 3, 2001.

[2] "By soumyanilcsc - own work, cc by-sa 4.0„" https://commons.wikimedia.org/w/index.php?curid=37353395, accessed: 07-12-2019.

[3] "Inra person dataset," http://pascal.inrialpes.fr/data/human/, accessed: 07-12-2019.