

People detection Laboratory

Session1 (session1.m script)

1

test_ReadGTBlobs

The script goes through the video tud-campus-sequence by frames and stores the number of images in the variable num_images and with the help of ReadGTBlobs detects Blobs. After with a for loop and a help of the recursive PaintBlob mark the blob with a green rectangle. The final image with the detected blob is then shown and paused for 0,001 secundum.

ReadGTBlobs From tud-campus-sequencegt.txt reads the basic parameters of the blobs, like the x, y parameters and score. Set the x1 and y1 to be smaller and not smaller than 0 and the weight and height of the blob from the difference of x1-x2 and y1-y2. Finally it removes the blobs that are outside of the process mask.

PaintBlobs The given parameters are the blobs detected by the read ReadGTBlobs, the image and the color green. With a food loop it goes through the second dimension of the blobs and calls itself recursively with the blob under the given index in blobs.

The blobs are stored in the txt files by 5 parameters. The first 4 is for the position of the surrounding rectangle and the 5th is the score. In the output image it is marked with a green rectangle. The data structure is a 1x71 cell for the images. For each frame there is a cell with a 1x6 struct where all 6 blobs have values: x, y, w, h, num_frame and score.

2

demo:

It loads the mat and jpg files. With the help of the function called test the process is run and between every part the running is paused and it waits for a key to be pressed to continue. After it displays the images loaded from the jpg file and the model from the mat file. The next image is made for detecting objects. With the function imgdetect the model is used on the image and boxes are marked in the displayed image. The last image is made using non-maximum suppression on the previously detected objects and for that the overlap value is 0.5.

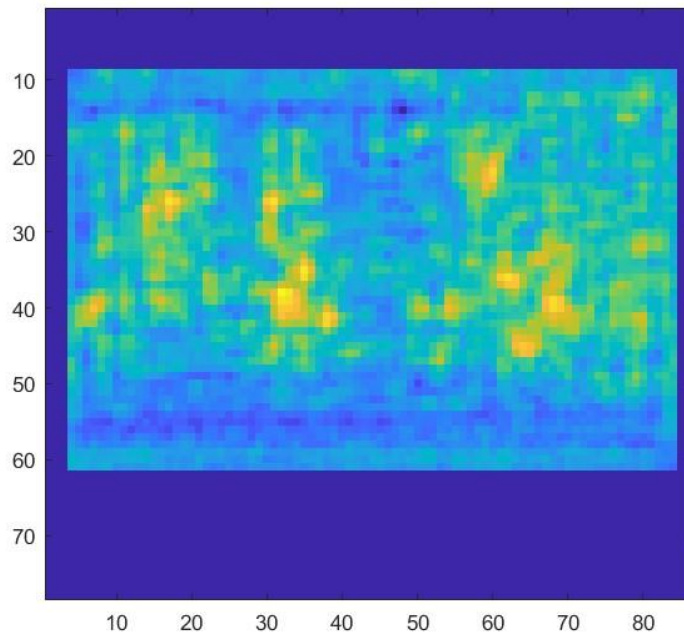
The most noticeable difference is that the paper is using 5 while the code 8 body parts. The coarse root filter and several higher resolution part filters detect more areas in the middle of the human body in the paper. In the final figure in the paper there are less features detected with blue and consequently in the code there is one more person detected incorrectly.

Person model:

- Identify the number of body parts. 8
- Identify the number of poses supported (mixture models).

`model.rules{model.start}` is a struct array of 2 elements so the mixture model has two components. In the paper one model is supported for the pedestrian picture.

- The Image and Feature pyramid in Figure 3.
The Feature pyramid is generated in the function called `imgdetect` with the help of `featpyramid` and stored in the variable `pyra`. The image is loaded from file and stored in the `im` variable.
 - The number of images in the pyramid or scales is 8 (`sbin = 8`)
- Identify where the final combined score is (see figure 4).
It is stored in `bbox` and then with `clipboxes` the ones that are outside are deleted.
- Visualize (with `imshow`) the final combined score at scale (level) 11.



- The Non-Maximum Suppression is made by the `nms` function (`top = nms(bbox, 0.5)`). It selects the high-scoring detections and ignores the ones that are significantly covered by the previously selected detection and this significance is 0.5 in this code.

3

test_DTDP_detector

The script gets the models name and goes through the video `tud-campus-sequence`, stores the

number of images and the image names. After the file is closed, it calls the DTDP_detector function.

DTDP_detector

It loads the model, goes through the images and calls the test function for each. In the test function calls imdetect for each image and with 0.5 overlapping value naked a non-maximum suppression. If there is detection after the nms, it created bboxes with bboxpred_get and clip it and nnms it and store them in te final_blobs variable. In the end it created the idl file with the detections.

The blob output format is quite the same as in the input base file. The path for the image then for all blobs in that frame the four x1, x2, y1, y2 position values and after the calculated scores.

4

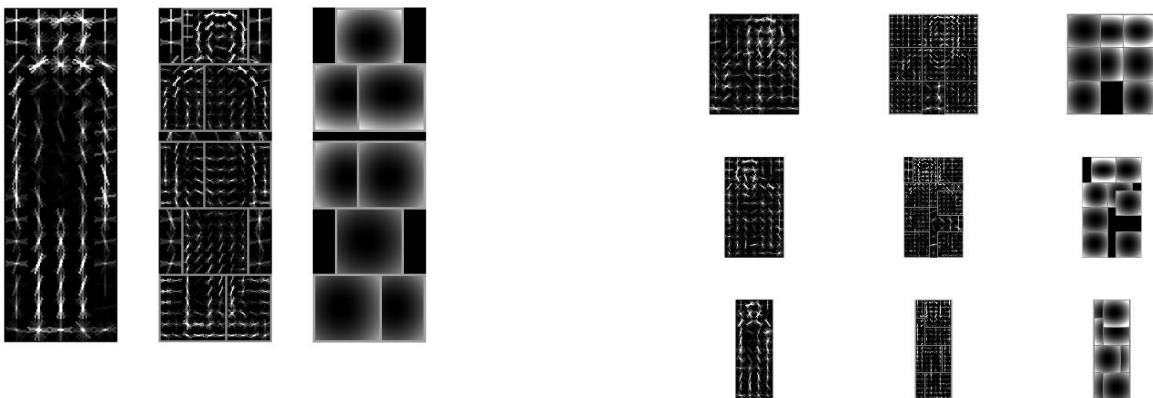
test_ReadBlobs

The script gets the models name and goes through the video tud-campus-sequence, stores the number of images and the image names. Call the ReadBlobs and Blobs_Threshold function. In every frame then the PaintBlobs marks the blobs with rectangles.

By selecting a larger debug_threshold value there are less and less false positive detections. With -0.8 it detects quite well.

With this function the blobs are not saved in a separate file, they are marked in the pngs and saved in a cell data structure.

5

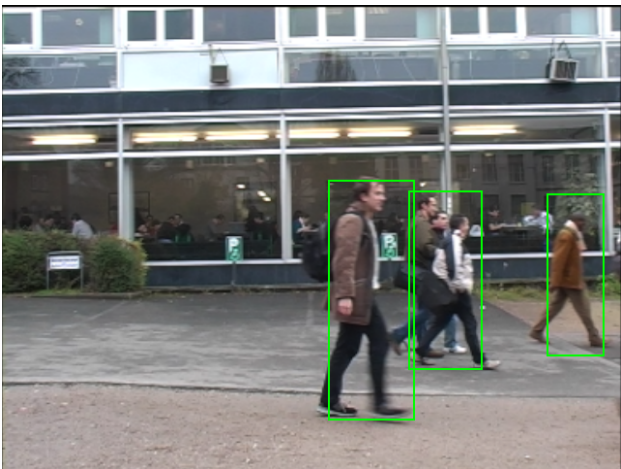
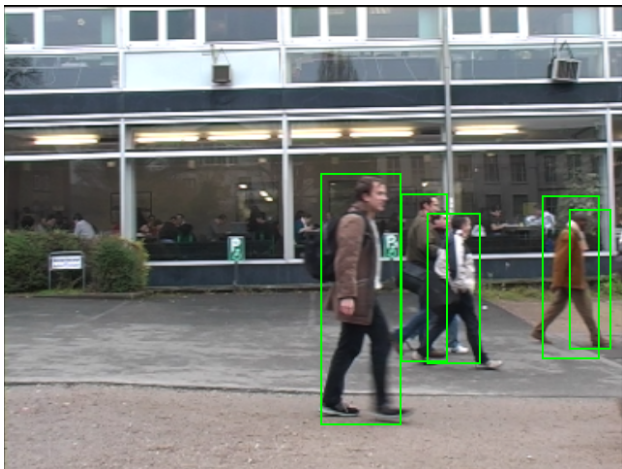


Compared to Inria Voc2009 was more precise with the pedestrians but detected falsely in one place. To get the same result with the videos instead of -0.8 debug_threshold I had to use -0.5 as it had

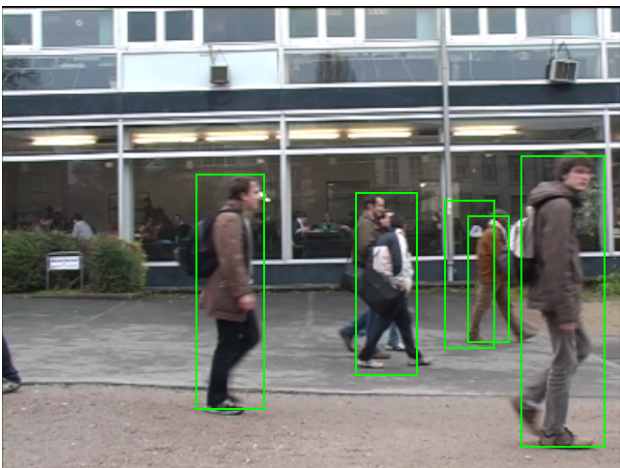
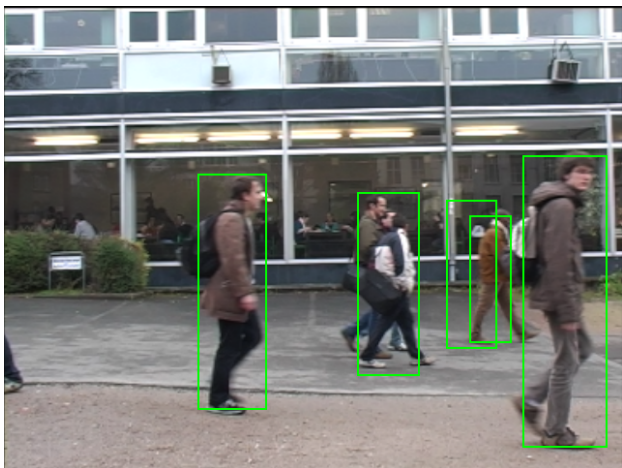
more falsely detected elements, but with the correct threshold it was able to detect one person once and more accurately.

Inria

VOC2009



tud-campus-sequence-006_out



tud-campus-sequence-055_out