

Performance of Cascade Classifiers used for open hand detection

ADITYA JAIN
PALLAB K. GANGULY
SAURADIP NAG
SWATI G. HAZRA

Kalyani Government Engineering College

February 1, 2018

Abstract

Following our last report, where we pointed out that smaller window sizes were more promising in terms of accuracy, we proceeded to test classifiers of smaller window sizes, with varying number of stages to obtain a working classifier model. This report deals with the description of the evaluation performed, its results and interpretation.

I. BRIEF DESCRIPTION OF TRAINING OF CLASSIFIERS

From our last series of tests, it was found that with the increase in window size, the number of positive samples successfully detected from a set of 200 images decreased. We therefore proceeded with smaller window size. In particular, we trained two classifiers corresponding to three window sizes:

- C1: Trained on 20x20 positive images
- C2: Trained on 30x30 positive images
- C3: Trained on 40x40 positive images

In addition to this a 50x50 classifier was used for benchmarking performance. Each of the above HAAR type cascade classifier was trained for 15 stages, using GAB algorithm.

II. TEST DATASET AND PERFORMANCE METRIC

The test dataset was created as below:

1. Raw positive images of hands (both L and R) grayscale, resized to 20x20

2. Superimposed¹ on random 100x100 grayscale images, created 500 superimposed images.

We also point out to the reader that the in dataset described above, each image consists exactly one positive image. Thus there are 500 positive images in our dataset. The location of the positives, and the height and width are available in the filename, as serve as a label for the dataset.

Once again, we used accuracy as the performance metric, as the dataset was not unfairly skewed. We define accuracy as the fraction of positive samples correctly detected from the actual number of positives in the dataset. From our previous experience, the optimum threshold a detected region has to cover against the actual region in the image is in the range 70-80 percent. Therefore, we did not vary the threshold in our present test.

III. PARAMETERS VARIED

The parameters varied and their ranges are described in Table 1.

¹Using opencv_createsamples

Table 1: Parameters

No.	Parameter	Range
1	Scale Factor	[1.01, 1.02, ..., 2.2]
2	Min. Neighbors	[1, 2, ..., 50]
3	Window Size	[20, 30, 40, 50]

Table 2: Algorithm

1. size \leftarrow 500
2. params \leftarrow scale, neighbours, stages
3. vals \leftarrow range of values of params
4. load cascade file
5. for each val in vals do:
6. count \leftarrow 0
7. for each img in training-set do:
8. detectMultiScale(img, params)
9. get predicted region for img, compare with actual labels
10. if predicted region covers $\geq 70\%$ actual region, count it as hit.
11. accuracy=hits/size
12. plot accuracy vs. params.

IV. ALGORITHM

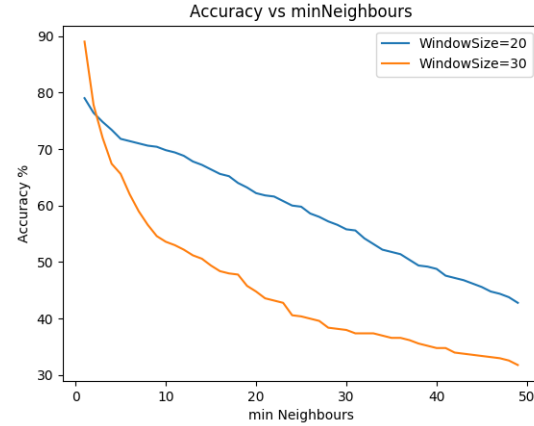
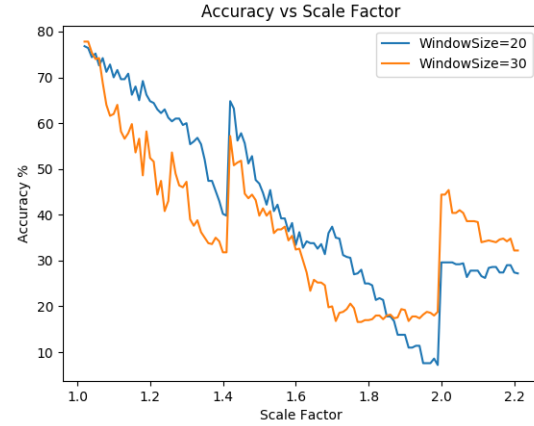
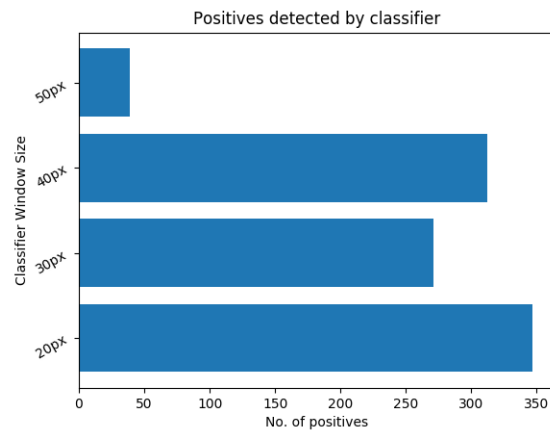
The algorithm is described in Table 2. It is virtually the same algorithm as used earlier.

V. RESULTS

Figure 1. depicts the variation of accuracy with neighbours, Figure 2. depicts the variation of accuracy with scale. Figure 3 represents the number of positives detected for different stages of classifiers.

VI. CONCLUSION

1. Based on the above evidence, the 20x20 classifier, for 15 stages seems to be the most promising candidate for future work.
2. We also may say that the values of scale factor and min neighbours should be in the range 1.01-1.05 and 2-10 respectively to achieve high accuracy.

**Figure 1: Accuracy vs. minNeighbours, scale=1.03, 15 stage classifiers****Figure 2: Accuracy vs. scaleFactor, minNeighbours=2, 15 stage classifiers****Figure 3: Positives detected vs. windowSize, scale=1.08, minNeighbours=9, 15 stage classifiers**