ECEN 649 Course Project (100 + 20 points)

Implementing Viola-Jones Algorithm

Kun Yang

November 2019

1 Introduction:

Viola-Jones is one of the most famous algorithms that is based on the original Ada-Boost thought. It has a great success in face detection before Deep Neural Network becomes the most popular one, and it is still used in all kinds of digital cameras. For our project, please go and check the Original Paper first.

In this project, we will offer you the dataset from CMU containing both face and non-face figures that are uniformly sized, and you will be given training set with 500 faces and 2000 non-faces picture, testing set with 472 faces and 2000 non-faces. Your responsibility is to utilize the dataset and implement the algorithm step by step. You should submit a detailed report after implementing the algorithm.

For this project, you can use any coding languages that are comfortable to you, and you can use all packages that help you compute the result and speed up your code. Still, you are not allowed to use packages that already have part or even full algorithm implemented inside (You can use them as your test or evaluation of your implementation).

2 Project Missions:

After reading the paper, you will notice that the whole algorithm is mainly divided into three parts, Extracting Haar Features; Training to find the best K Haar Features; Use Ada-Boost to achieve the detector. Our project will also be judged in these three aspects.

2.1 Extract Haar Features (20 Points):

The Haar features are five different kinds of filters with all possible sizes that are applied to the system. Please pay attention to how they compute the features, use the integral image will save you a massive amount of work. The extracted features should be a vector with a specific size, carefully calculate how many Haar features you should generate for each picture.

In your report you should include:

- The total number of Haar Features is: *****.
- There are **** type 1 (two vertical) features.
- There are **** type 2 (two horizontal) features.
- There are **** type 3 (three horizontal) features.
- There are **** type 4 (three vertical) features.
- There are **** type 5 (four) features.

Notice that the pictures in our dataset are (19 * 19) instead of (24 * 24), so the total number of features is much smaller than the original paper. Also, the successfully implement this part, I recommend you to restrict the size of the filters first (for example, limit the max-width of the filter to 8 as well as the max-length).

2.2 Build Your Adaboost Detector (50 Points):

After extracting the features, employ the AdaBoost algorithm and find the detector with 1, 3, 5, 10 rounds. For each different detector, you need to show the feature you choose, the threshold you have, and at last, draw your top one feature for each detector on a test image (like the original paper).

In your report, you need to include:

- Feature number 1:

Type: Two Horizontal

Position: (10,10)

Width: 8

Length: 4

Threshold: 0.5555

Training accuracy: 0.61

The pictures your plot should look like this 1:

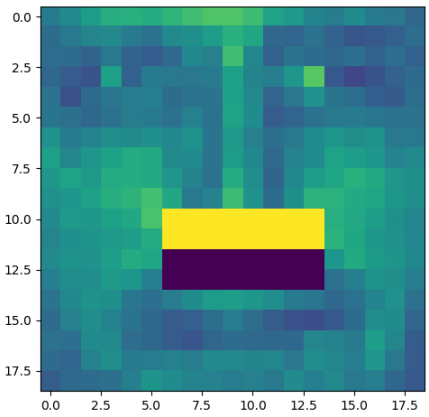

Figure 1: Top 1 feature for 3 rounds Adaboost

Also, as your rounds of boosting grow, find out how will the accuracy, false-positive rate, and false-negative rate change? Plot this change, explain why this would happen.

In your report, you need to include:

- Adaboost rounds: 1

  Total accuracy: 0.55 (550/1000)

  False Positive: 0.2 (200/1000)

  False Negative: 0.25 (250/1000)

As for the plot, you can draw whatever plots you want as long as it can help you explain why the system changes in this way as you add more features inside.

2.3 Adjust the threshold (30 Points):

In the real world, there are different standards for the face detection system. We may want to eliminate as much false alarm as possible in daily life, for example, we can tolerate not being recognized as a face but cannot tolerate the whole environment are all recognized as faces. However, if you build the system for security reasons, you don't want to miss any of the suspicious moves.

In order to balance these two kinds of losses, we need to use different criteria to train your detector, for example, use false positive instead of the empirical error. Your mission is to train different 5 round AdaBoost detectors, show us how their false positive and false negative changes. What change did you make on your system? Please use a table to summarize your work.

Criterion	Total Accuracy	False Positive	False Negative
Empirical Error	61%	19%	20%
False Positive	50%	10%	40%
False Negative	59%	30%	11%

Table 1: An example of the desired table

2.4 Additional Part (Optional, 20 Points):

There are two different topics you can choose. You only need to choose one to get these 20 points.

2.4.1 Build the cascading system:

Use the cascading method in our class, train the detector up to 40 rounds, and compare it with the single AdaBoost detector. Does your result get better? Does the training get faster? Compare your results with what you have got in our baseline part.

In your report, you need to show me after passing each detector in your cascading system, how many non-faces photos have you abandoned? What is the final accuracy of this cascaded system? Can you explain why this improvement/fallback happens?

2.4.2 Detecting in real-world photos:

Build a classifier to do face detection on the colored photos with different sizes. Also, you need to use the cascade method we talked about in the class, and you may also need to do some pre-processing on the target photos.

You need to submit the processed photo with all faces detected have a rectangle frame around it. Describe your system, how you designed it, and how it works.

3 Report Requirement:

To write a good report, you need to include all the required lists, graphs and tables in it.

As for the codes. If you are familiar with GitHub or other similar tools, please give me a link to your project. If not, please pack your source code together with your report. You also need to write a Readme file include the following information (Notice, if your didn't include the Readme file and we can not run your code properly, you are in trouble),

- What kind of external packages you used, for what purpose?
- How to run your code (include getting the environment ready).

The score of the project will not only depend on your result; the analysis and understanding of the algorithm are also critical. Don't be panic if you can not get the right answer, but copying other's code is not acceptable.

Good luck with your final project! Feel free to ask me any questions if you have any problems with the project!