**INSTRUCTIONS**

1. Create a folder with the negative images.
2. Create a folder with the positive images.
3. Run the program ‘createnegativesfile.py’. It will ask you to enter the absolute path of the folder with negatives images. The output is a text file with the paths to each image in the negatives folder.
4. Run the program ‘resizepositives.py’. It will ask you to enter the absolute path to the folder with the positives images. This program will resize and convert the images to gray.
5. Run the program ‘objectselection.py’. It will ask you to enter the absolute path to the folder with the positives images as in the step before. The first image will be shown. You have to select the object to detect, text in this case, in the image using you mouse. Press when you want to start the rectangle and keep pressing while you draw a rectangle around the text and then stop pressing. Be aware that the rectangle will not be shown until you stop pressing but it is being created. When it is shown press the key ‘c’ to confirm the selections and then the key ‘s’ to save it. The next image in the positive image folder will be shown. Repeat this until no image is shown what will mean that all have been shown and the process is done.
6. Open the program called ‘HaarTraining.py’ and change the paths in the ‘main’ program to those you are using. You need to give the paths for the positives text file generated from the ‘objectselection.py’ program, the text file generated by the ‘createnegativesfile.py’ program, the vecfile\_path should be path where you want the .vec file to be created and the output\_dir path is the one to an empty folder you have to create to store the output cascade files.  
   You should also modify the parameters under #vecfile:

* num\_pos and put the number of positive images you have
* w and h to the width and height you want the classifier to find objects

You can modify as well the parameters under #train cascade variables:

* num\_pos2: it is the number of positive images to be used every classifier stage, it should be a number smaller than num\_pos.
* num\_neg: it should be the number of negative images you have.
* num\_stages: it is the number of stages you want to have.
* bufferSize: it is how much RAM memory you want to be used by the process.
* featuretype: it can be HAAR or LBP.
* minHitRate: it is the percentage of positive identification you want to achieve. It should be big really close to 1 and as closest it is the longer it will take the program to run.
* maxFalseAlarm: it is how many samples you allow to be wrong classified.

Once you have modified these parameters to the ones you want you can run the program. Be aware that it may raise an error like this: “Traincascade Error: Bad argument (Can not get new positive sample. The most possible reason is insufficient count of samples in given vec-file)”. It is not caused by the program but by the num\_pos2 parameter, you need to reduce it more in comparison to the num\_pos.

1. Open the program ‘detector.py’ and put the name of you classifier file .xml inside the function cv2.CascadeClassifier. The run it. It will ask you to enter an image filename and it will display it. Press the key ‘d’ to detect the text in the image or the object you have trained the classifier for.