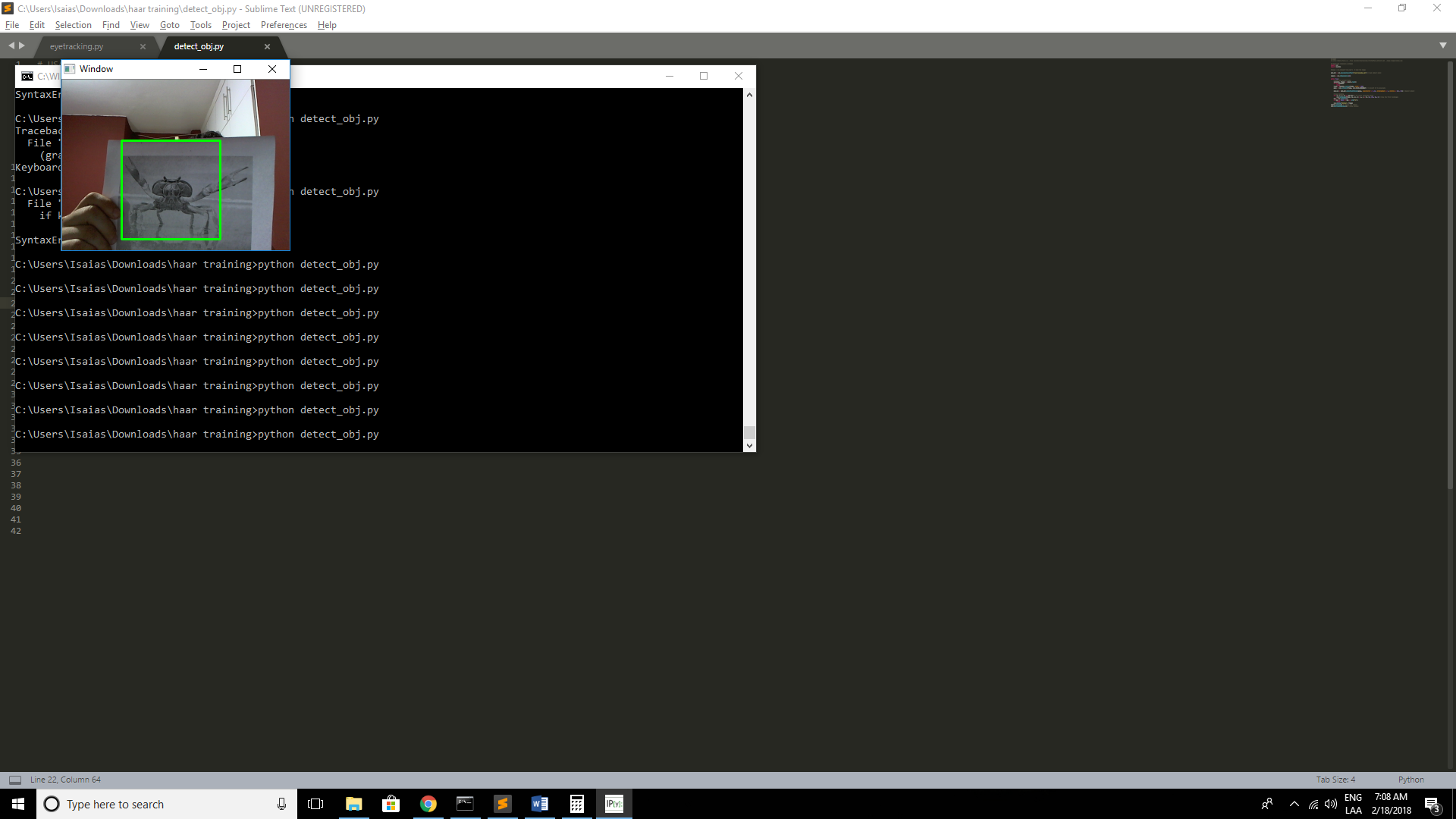
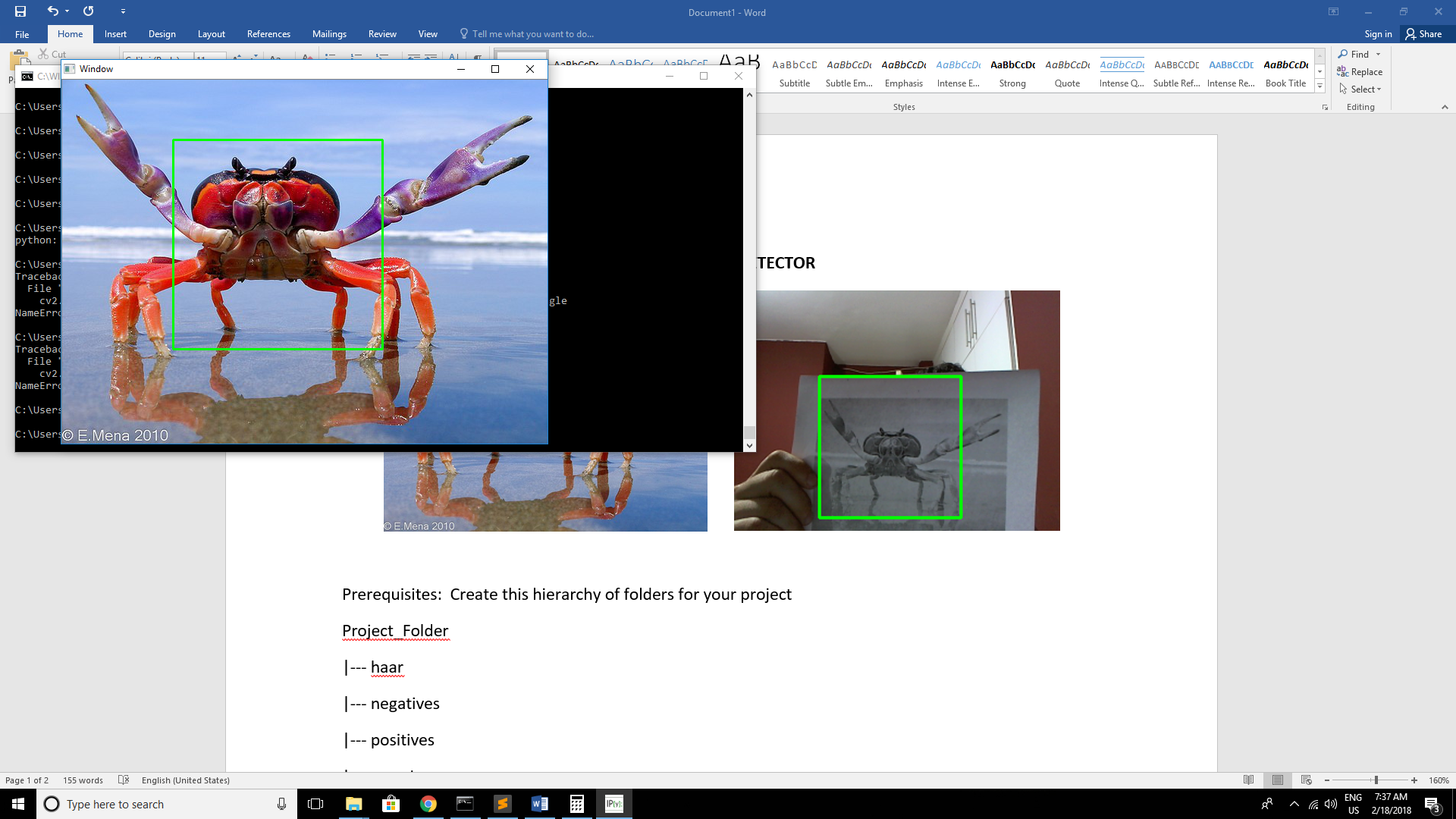
**TRAIN A CRAB DETECTOR**





Prerequisites: Create this hierarchy of folders for your project

Project\_Folder

|--- haar

|--- negatives

|--- positives

|--- samples

|--- createTrainData.py

|--- detect\_obj\_image.py

|--- detect\_obj\_video.py

|--- download\_images.py

|--- imutils.py

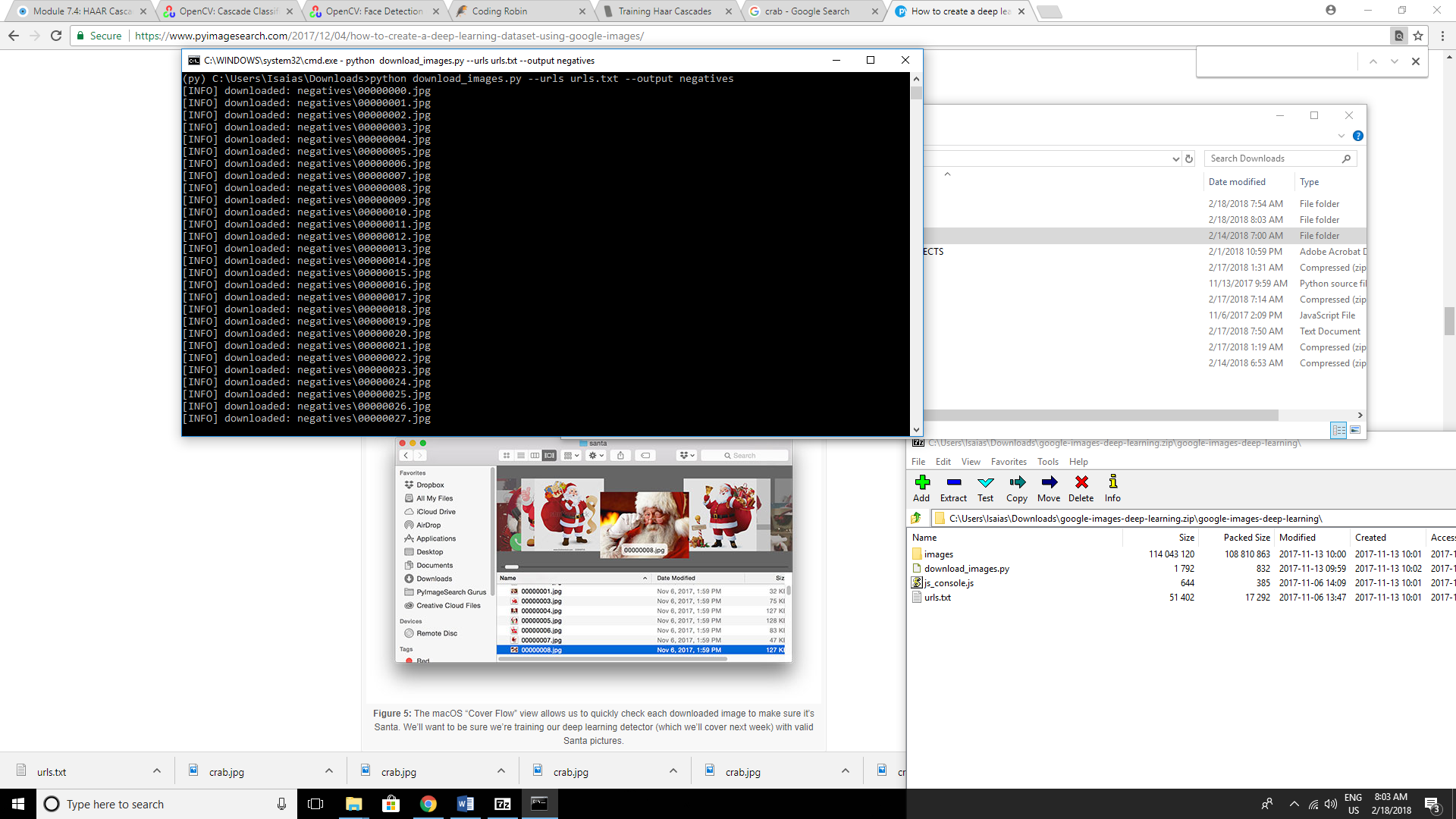
|--- js\_console.js

Folders haar, negatives, positives, samples are empty for the first time use

Training Steps for Haar Cascade Trainer

Step 1: Collect positive and negative images

* Positive images in this case were collected from 101\_ObjectCategories.tar.gz file
* Folder used is 101\_ObjectCategories.tar\101\_ObjectCategories\crab\
* Put the images in the positives folder
* For negative images we searched google images and used Adrian Rosebrock (pyimagesearch) code for image download, see references.
  + Search for google images
  + Scroll down until all images you want load
  + Press Shift+Control+I
  + Click on Control tab
  + Open the js\_console.js file and…
  + Copy the first paragraph to the google console tab and press Enter
  + Copy the second paragraph to the google console tab and press Enter
  + Copy the third paragraph to the google console tab and press Enter
  + The generated urls.txt, place it on the project folder
  + Run ‘python download\_images.py --urls urls.txt --output negatives’ for the negatives



Step 2: Generate positives.txt and negatives.txt file

* Run this on the command prompt inside the positives folder ‘dir /b \*.jpg > positives.txt’
* Put this file on the project folder and edit the inside data as shown below

positives/image\_0001.jpg

positives/image\_0002.jpg

positives/image\_0003.jpg

positives/image\_0004.jpg

positives/image\_0005.jpg

positives/image\_0006.jpg

positives/image\_0007.jpg

…

* Run this on the command prompt inside the negatives folder ‘dir /b \*.jpg > negatives.txt’
* Put this file on the project folder and edit the inside data as shown below

negatives/00000000.jpg

negatives/00000001.jpg

negatives/00000002.jpg

negatives/00000003.jpg

negatives/00000004.jpg

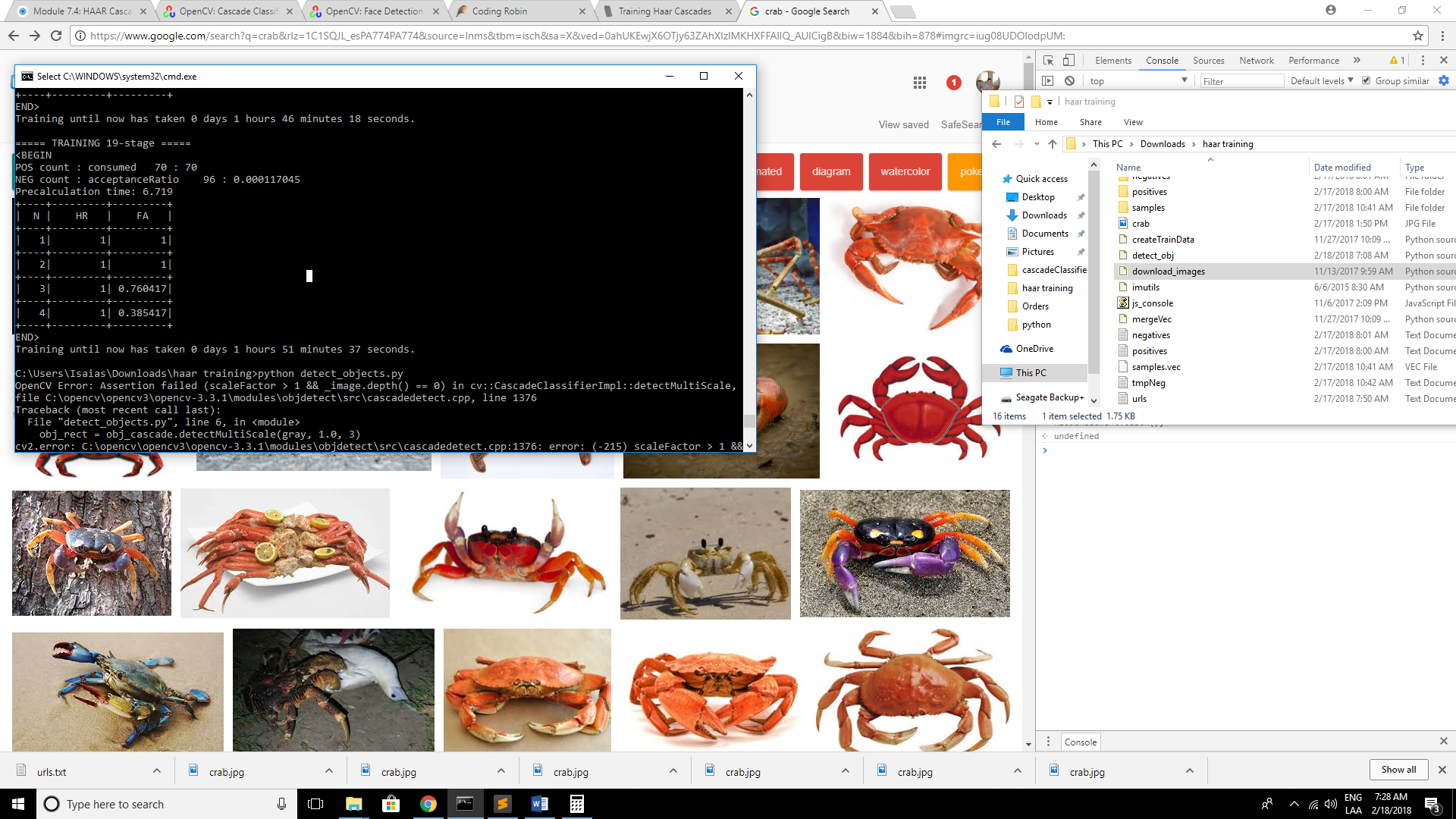
negatives/00000005.jpg

negatives/00000006.jpg

…

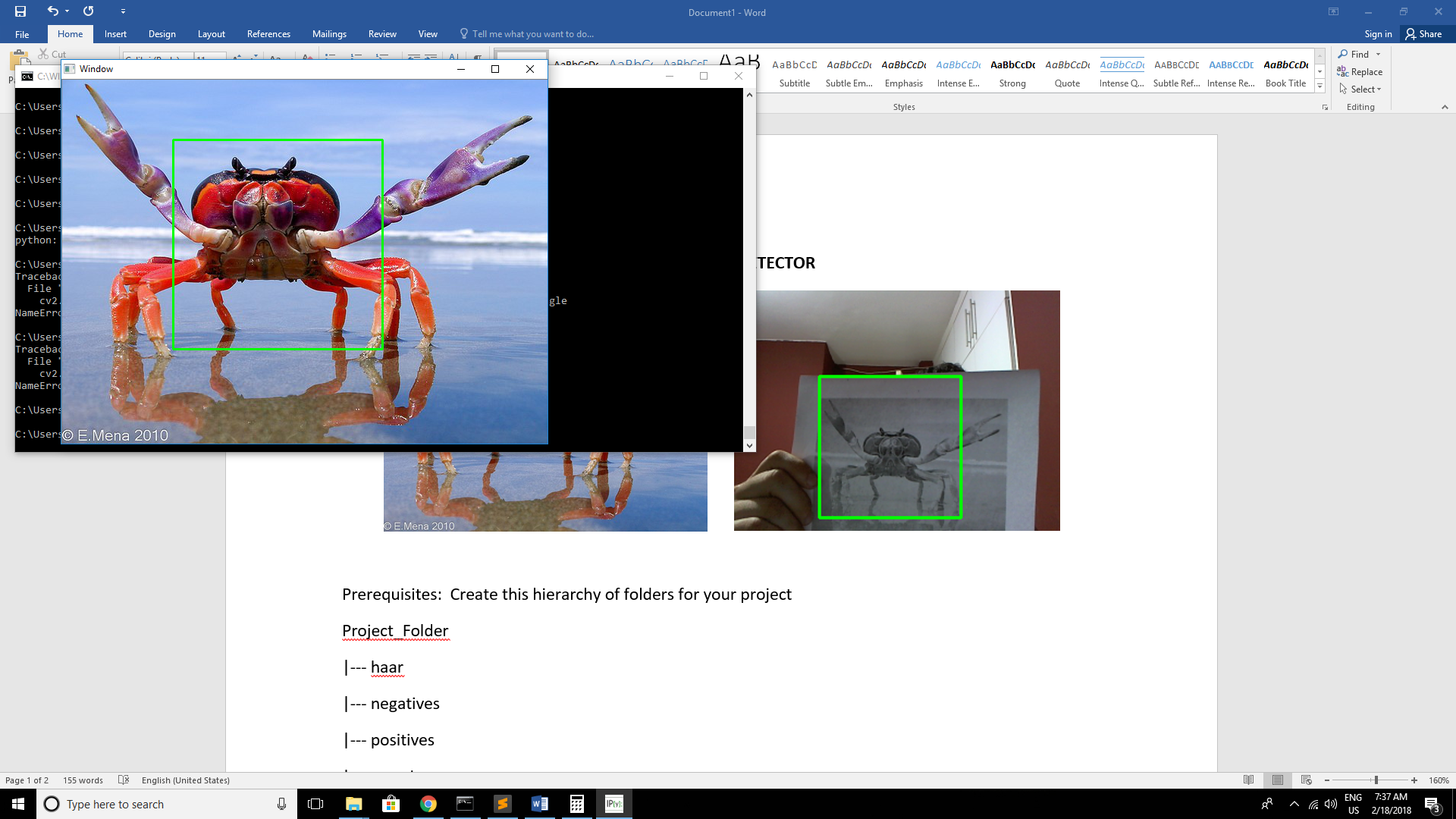
Step 3: Create samples.vec, vector.vec file and train your classifier

* Run this command on the project folder ‘python createTrainData.py positives.txt negatives.txt samples 70 "opencv\_createsamples -bgcolor 0 -bgthresh 0 -maxxangle 1.1 -maxyangle 1.1 maxzangle 0.5 -maxidev 40 -w 60 -h 60"’
* In my case I only had 70 samples, that’s not enough data for a good haar cascade, you must collect more data, for the parameters –w and –h you must preserve dimensionality.
* Generate de vectors.vec file running this command ‘python mergeVec.py -v samples/ -o samples.vec’
* Run this command on the project folder ‘opencv\_traincascade -data haar -vec samples.vec -bg negatives.txt -numStages 20 -minHitRate 0.999 -maxFalseAlarmRate 0.5 -numPos 70 -numNeg 96 -w 60 -h 60 -mode ALL -precalcValBufSize 1024 -precalcIdxBufSize 1024’
* You must match numPos and numNeg from your number of images of each folder (positives and negatives folder), -w and -h must be the same as the first step.
* Depending on the training data you can see that it should take minutes, hours or days to train. A good model is trained between several hours, maybe more than 10. Below is a fast trained model just for demonstration purposes.

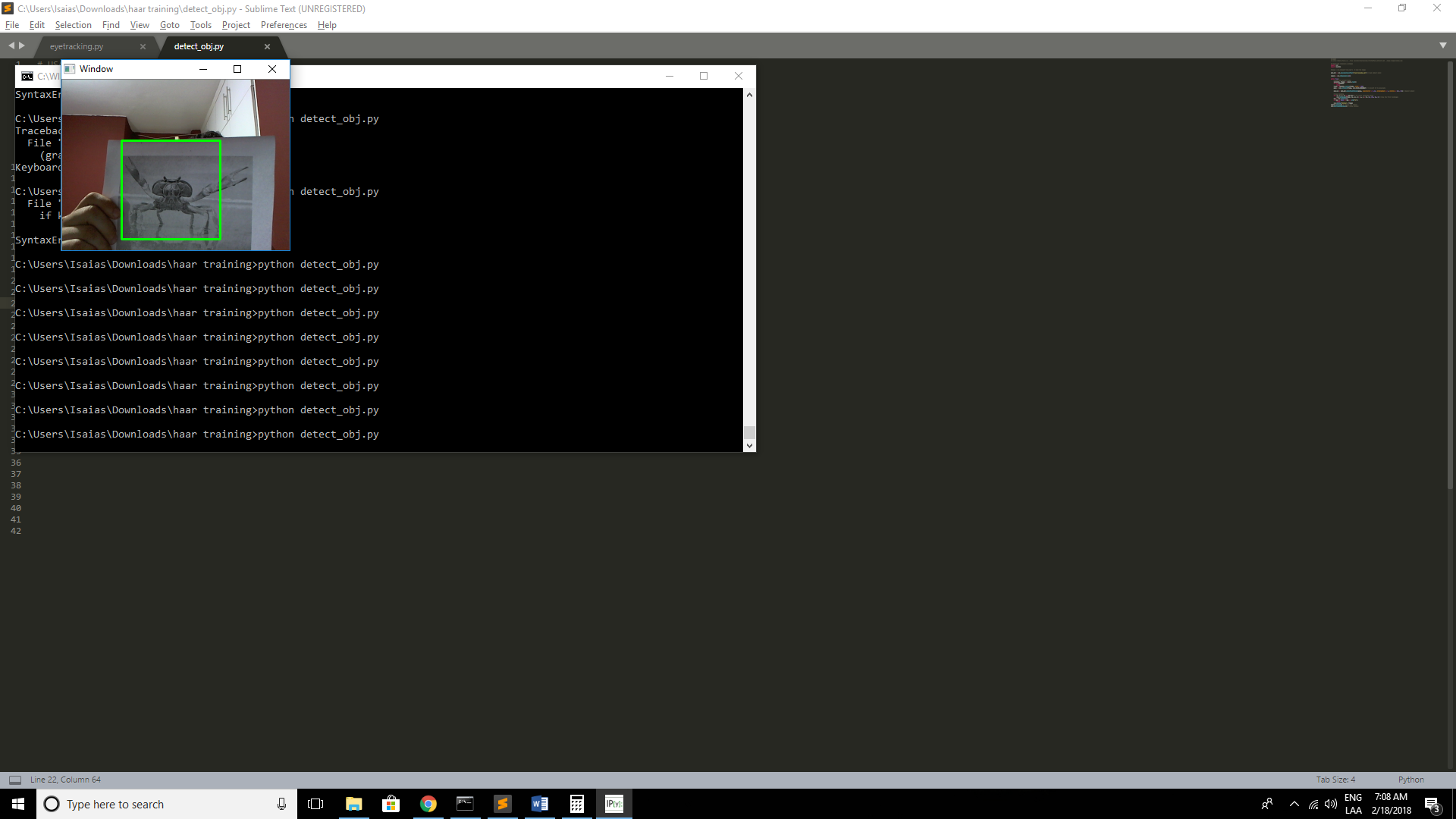


Step 4: Run your classifier

* The model is in the haar folder and its called cascades.xml
* For test over an image just run ‘python detect\_obj\_image.py’ this will load the ‘crab.jpg’ and try to locate the crab.



* For test over a video just run ‘python detect\_obj\_video.py’ this will try to do real-time detection.



References

<http://www.vision.caltech.edu/Image_Datasets/Caltech101/>

<https://www.pyimagesearch.com/2017/12/04/how-to-create-a-deep-learning-dataset-using-google-images/>

<https://docs.opencv.org/trunk/dc/d88/tutorial_traincascade.html>

<http://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html>

<https://memememememememe.me/post/training-haar-cascades/>

<https://www.learnopencv.com/training-better-haar-lbp-cascade-eye-detector-opencv/>

Goals

* Do the same thing as above, BUT, trained with images of Liam Neeson (50%)
* Train using other classifier for (see table below) (25%)
* Port the new trained model to the raspberry pi (video) (25%)
  + The idea is to present a photo of your trained object and it must be recognized
  + Tun your parameters for better responsiveness

|  |  |  |
| --- | --- | --- |
| ID | Nombre | Train Object |
| 1 | AGUILAR ACEVEDO GABRIEL EDUARDO | Camaras |
| 2 | BELFORD PETERS SHANTIEL AISHA | Cocodrilos |
| 3 | CAMPOS MONTOYA KADIR | Dalmatas |
| 4 | CASTILLO ARAUZ ELIECER VALENTIN | Elefantes |
| 5 | CHUNG ZOU ERNESTO | Camellos |
| 6 | FERMAN GONZALEZ ADRIAN ANTONIO | Canguros |
| 7 | GONZALEZ BONILLA RODNEY ALEX | Motocicletas |
| 8 | MEJIAS STABILITO JUAN LUIS | Carros |
| 9 | MENDOZA LEIVA ALDAIR ANTONIO | Palomas |
| 10 | MORENO URRIOLA ROGELIO ROBERTO | Rinocerontes |
| 11 | PEÑA ABREGO ALCIDES | Silla de Ruedas |
| 12 | QIU CHEN MICHAEL | Revolver |
| 13 | ROMERO RAMIREZ PEDRO ENRIQUE | Pizza |
| 14 | RUIZ CORONADO DYLAND CHRISTIAN | Langostas |
| 15 | SALCEDO BORJA DANIELA | Leopardos |
| 16 | TELLEZ PEREZ PAOLO RICARDO | Flamencos |
| 17 | VERGARA PAREDES MIGUEL ANGEL | Puercoespines |
| 18 | ZAMBRANO VIDAL CESAR AUGUSTO | Laptops |
| 19 | CHRISTIAN CRUZ | Escorpiones |