Practice 2

David García and Héctor Aparicio

I. INTRODUCTION

In this practice we are going to classify some objects on video using the code of a repository on github. The practice is divided in two sessions. First session we download the code, understand how to implement and create a project that allows other people run it. On session 2 we take this code and we are going to use two methods of object detection. Visually we will compare and decide wich is better.

II. METHOD

For the first session we use a repository on GitHub. We have to understand it and execute exactly what the readme text say. It's not complete so the first part is understand how to do it. With that we can do our own repository in wich we will explain exactly how to run the code for object detection.

In session 2 we use the code for do a object detection task on some of our videos. We compare 2 methods.

A. YoloV2 + SeqNms

This one is used on the first session. Use 2 techniques for object detection. First one is the Yolo method. It detect and classify on the same step. First divide the frame on grids and every grid have a classification associated. This makes a lower computational cost. Then apply the second method known as SeqNms. This one use a threshold and if the number of the detection on the frame is higher take that as a final result. Then add every detection on frames and classify the object. The detections that are under of the threshold are eliminated.

B. YoloV2

It's exactly the same method but without the post-processing method called SeqNms. Now, there's no threshold that make stronger the classification task.

Using this simple method and the post-processing one we have to compare it and visually determine wich is better.

III. IMPLEMENTATION

A. Repository code

First we have to create a project on GitHub. Put in private and call it Seq_nms_YOLO. Then create the repository and select without a Readme. Insert https://github.com/melodiepupu/seq_nms_yolo when ask you for a url. In this moment we have all the project created on our own GitHub. Now we go to the computer and select a path for make our python 2 virtual enviroment. We did it in the computer 21 of the laboratory, so we selected the personal carpet. For the creation of the enviroment u have to use a terminal and run "virtualenv -p python2 yolo_env", and then activate with "source yolo_env/bin/activate" and "source activate". Next step is install all the requeriments for the correct execution of the code. We can use this on terminal "conda install -y -c conda-forge opencv", "conda install -y matplotlib Pillow scipy tensorflow" and "conda install -y -c conda-forge tf_object_detection". Now we clone the repository on our root direction with "git clone https://github.com/hectordavideps/seq_nms_yolo.git". We are cloning from our own private GitHub.

Using the comand "cd" we are going to move inside our new directory called seq_nms. Inside there download the yolo and tiny weights using "wget https://pjreddie.com/media/files/yolo.weights" and "wget https://pjreddie.com/media/files/yolov2-tiny-voc.weights"

Now we are going to run the code. First we will do a makefile using the command make in our terminal. Then we select a random short video in our case we input.mp4 as an example and introduce it in the video folder. Inside this carpet we use "python video2img.py -i input.mp4" for separate the video in frames and then we use "python get_pkllist.py" for take this characteristics. Then we go on the terminal window to the root directory and hange code on yolo_detection.py and use this in line 37 lib = CDLL("xx/libdarknet.so", RTLD_GLOBAL) where xx is the root of your project. There we run "python yolo_seqnms.py" for use the neural network and make the classification of the frames. With this output images generates we go to our video folder and run "python img2video.py -i output" for make the video with all the frames classify. Now we got a output.mp4 with the answers.

B. YoloV2 Version

We have to do exactly the same thing than the section A. The only thing different is change 2 lines from the function dsnms of yolo_seqnms.py. We comment lines 205 and 206. This cancel the NMS post-processing.

IV. DATA

For session 1 we can use any short video. Input.mp4 is a theorical one.

In session 2 we used 5 videos from the UFC 101 database [1].

v_ApplyEyeMakeup_g19_c03.avi, v_ApplyLipstick_g21_c01.avi, v_Archery_g07_c01.avi, v_BabyCrawling_g11_c01.avi, v_BalanceBeam_g18_c01.avi.

V. RESULTS AND ANALYSIS

A. Session 2

We did a visual observation of both methods. The one with the threshold expect a lower amount of objects classify, and

the Yolo version expect more objects and incorrect answers.

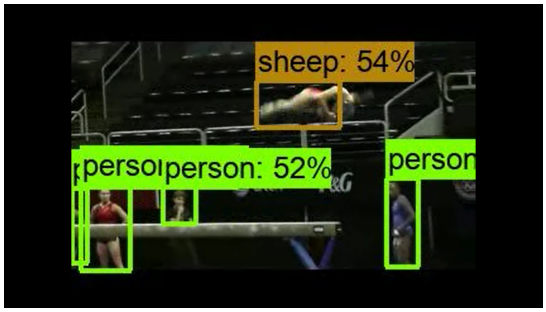

Fig. 1. vBalanceBeamg18c01.avi result

In the first figure we used the second method but something unexpected in this video is that both methods classify the girl doing stunts as a sheep. This mistake happen in both cases.

Fig. 2. vApplyLipstickg21c01.avi with NMS

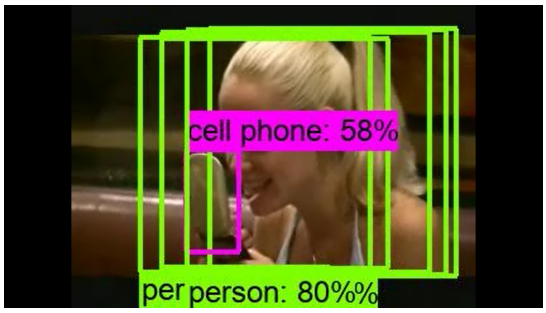

Fig. 3. vApplyLipstickg21c01.avi without NMS

Fig. 4. vApplyEyeMakeupg19c03.avi with NMS

Fig. 5. vApplyEyeMakeupg19c03.avi without NMS

There are 2 observations that we have to do. First one is the number of detections. When the NMS is desactivate, doesn't exist a threshold that says what is an object or not. So all the detection are available. In figure 5 we can see this. There are only 2 persons but the classifier detect 10 or more. The NMS version only classify 3 persons, the difference is huge.

Second thing is the mistakes. Wihtout NMS when the network have some similaritys with an object will classify that object. In figure 3 we can see both errors. Detect a lot of persons when is only 1 in the frame, and also say that a little mirror is a phone. This happen for the threshold reason. In figure 2, also detect that object as a cell phone but this detection don't pass this threshold and is eliminated. With the rest of the persons too, just take 1 detection of the person because is the most effective one.

VI. CONCLUSIONS

In session 1 we familiarize with GitHub. We saw that is important make a good Readme with all the steps because it's difficult use external code without a good explanation. We had to do some changes and understand the code for run it. Also the computers, environments and installations are not the same, so we learned how to adapt for our concrete case.

In session 2 we discover that the post-processing is a very strong tool. We saw that NMS version is huge better than without NMS. The Yolo version detect lot of objects and makes lot of mistakes but using NMS, we can improve it a lot.

VII. TIME LOG

	Task (h)	Memory (h)
Session 1	3	4
Session 2	2	3

TABLE I

TIME FOR EACH SESSION

REFERENCES

[1] Soomro K, Zamir A, Shah M. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild [Internet]. arXiv.org. 2012. Available from: https://arxiv.org/abs/1212.0402