Computer Vision 0510.6251

סמסטר א' תשפ"א

Project Instructions

1. The project is due in pairs on January 3, 2021. (03.01.21)
2. The project deals with the task of bus localization and classification (detection).

   Using the training images (from moodle), you will need to develop an algorithm that detects and classifies buses. At the end of the semester, each pair will be given **15 minutes** to explain briefly how the algorithm works and test its performance on new images that the algorithm hasn't seen.
3. The grades will be determined based on performance competition. The final score of the project will include **detection performance** score (details in Section 10) weighing 75% and **time performance** score weighing 25%:

   $$\textbf{score} = 0.75 \cdot \textit{Dice} + 0.25 \cdot (\textit{relative time})$$

   As mentioned, the final grade of the project will be determined by comparing the final score of all projects.
4. The project must be written in Python (v3.7).
5. It is recommended to install Anaconda (for python 3.7) and can be downloaded from the official website. To create a virtual environment with all the default packages, you can run the following command (in the anaconda prompt/cmd/terminal):

```
conda create -n <environment name> python=3.7 pytorch=1.3 tensorflow=1.15
torchvision matplotlib keras imageio
```

for managing the environment you may find helpful commands here.

6. It is highly recommended to prepare the project with deep learning tools, but it is not mandatory. You may use any of the leading deep learning packages: TensorFlow, PyTorch, Keras.
   Furthermore, you may use the Computer Vision Library: OpenCV, TorchVision (requires PyTorch).
7. The test will be performed on the Google Colab platform, using a GPU, so it is recommended to use GPU (both in the training and test phase). Without a GPU, the performance time may be highly affected.
8. The problem: Finding the location of each bus is in the image and identifying which bus is it (there are a total of 6 different buses). The detection of the bus will be done by the minimal blocking rectangle (bounding box) that contains the entire bus (examples below). The rectangle is defined by 4 parameters: xmin, ymin, width, height. Where xmin, ymin are the coordinates of the upper-left corner of the rectangle. In addition, a number between 1 and 6 will define the type of the bus, where {green = 1, yellow = 2, white = 3, gray = 4, blue = 5, red = 6}.

In summary, each detection has 5 parameters, the first four parameters define the location of the bounding box (localization) and the fifth parameter defines the bus type (classification). The output of the algorithm should be a text file where each row defines the annotations of the detections of one image (the parameters are comma-separated), as follows:

PIC.JPG:[xmin1,ymin1,width1,height1,color1],..,[xminN,yminN,widthN,heightN, colorN]

for example:

DSCF1013.JPG:[1217,1690,489,201,1],[1774,1619,475,224,2]

DSCF1015.JPG:[641,1342,1181,892,3]

DSCF1016.JPG:[1067,1843,1114,613,4],[1954,1278,1021,561,6]

In the above example, there are three images, in DSCF1013 and DSCF1016 two buses were detected and in DSCF1015 one bus was detected.

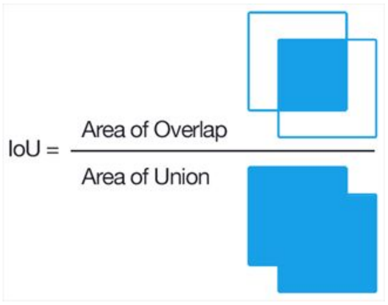
9. A file called ‘annotationsTrain.txt’ with ground truth tags (with the explained format) can be found in Moodle.
10. Correct detection definition:

Detection will be evaluated by the overlap ratio defined by IOU (intersection over union) between the detected rectangle and the ground truth rectangle (label). The overlap ratio is defined by:

$A - detection\ bounding\ box$

$B - ground\ truth\ box$

$$IoU = \frac{A \cap B}{A \cup B}$$



Detection will be considered as true positive (correct) provided that the following 2 conditions are met :

- IOU > 0.7
- The bus is classified correctly (correct color).

Finally, F1Score (Dice coefficient) is calculated. It is the only measure that weighs all the results of the detections from all images, it is the measure by which the quality of the project will be examined relative to other projects (which is 75% of the project’s grade).

11. Detection script

    You must edit the file ‘runMe.py’, the file contains a function called ‘run’ that receives two inputs: the file name of the annotations file that will be generated (explained in section 8) and the folder name in which the images that are to be tagged (validation\test images).

    That is, the definition of the function should be of the form:

    def run(estimatedAnnFileName, busDir):

    It is important to use the above names in order for the test script (explained below) to work smoothly.

    You need to write the function *run* to run your model.

12. The zip ‘python_scripts.rar’ from the moodle, contains 3 python files: Main.py, runMe.py, busProjectTest.py and Test_Project.ipynb.

    runMe - this file generates the annotations file. Currently, it is done by adding noise to the original GT file (the ground truth annotations). You can run it to check that the python is working properly (the GT file should be in the folder from which the script is called). You will need to replace this file with your file that prepares the estimated annotations file.

    busProjectTest - this script compares between the GT annotations to the estimated annotations. This file should not be changed.

    Main - the only script you will need to run, this script tests your project on the testing dataset. It is most important to check your algorithm with this script. This file should not be changed.

    Test_Project - an auxiliary script for testing the project in Google Colab.

    To check that the testing script is running properly, download all 3 files to a single folder and run **Main**. It will call the 2 other files, and calculate for each image from the given buses directory, the amount of true positive (TP), missed and false positive (FP) detections, generates the final F1 score, and saves the images (with the estimated bounding boxes) to the given output directory.

running **Main**:

Main receives as arguments, the GT annotations file (which can be downloaded from the moodle), the estimated annotations file name (that runMe creates), the directory with all the input images, and the directory that will store all output images and results. Before each input name it is required to state which input it is:

-myAnns <estimated annotations file>

-anns <real annotations file>

-buses <directory of all the input images>

-saveDir <directory for the output of the script>

You can run this script in the command line, example:

python Main.py -myAnns *newAnns.txt* ***-anns*** *realAnns.txt* ***-buses*** *busesDir* ***-saveDir*** *dirToSave*

In addition, for an explanation of each input, you can run the script with -h. That is:

python Main.py –h

Note that your projects will be tested on Google Colab, so make sure you are able to run the Main script from Google Colab.

To do this, you need to upload your project files and the Test_Project.ipynb to a Google Drive Folder, Open Test_Project.ipynb with Google Colab and run the script.

Make sure you use GPU in Google Colab: Tutorial for using GPU in Google Colab

13. Project submission:

    Each pair must submit until the scheduled date for the project submission:

- All of the files required to run the detection algorithm, and a 2-3 page report explaining the implemented solution (pdf file) should be submitted to Moodle as a zip file called: cv_proj_21_ID1_ID2.zip

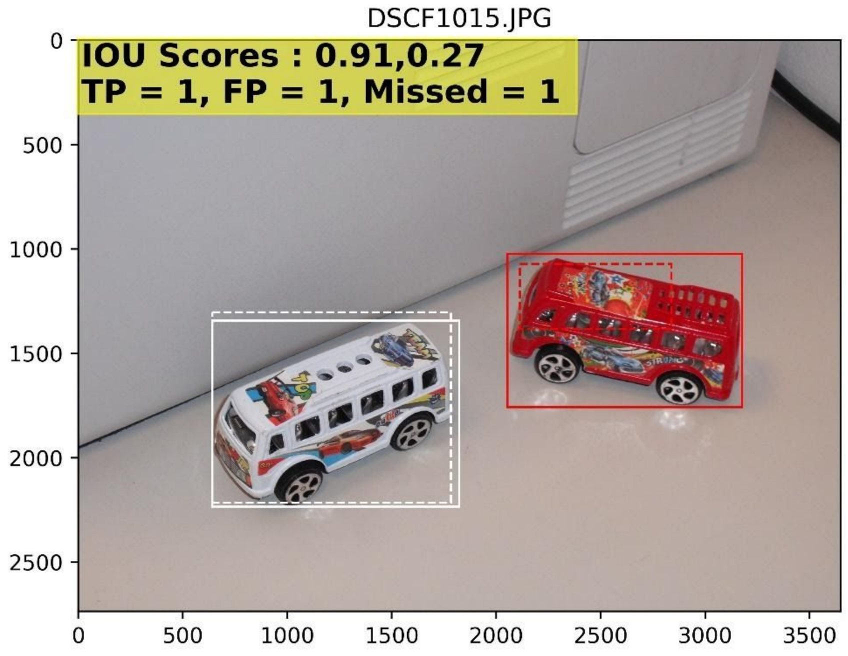
  ID1, ID2 should be replaced with your IDs

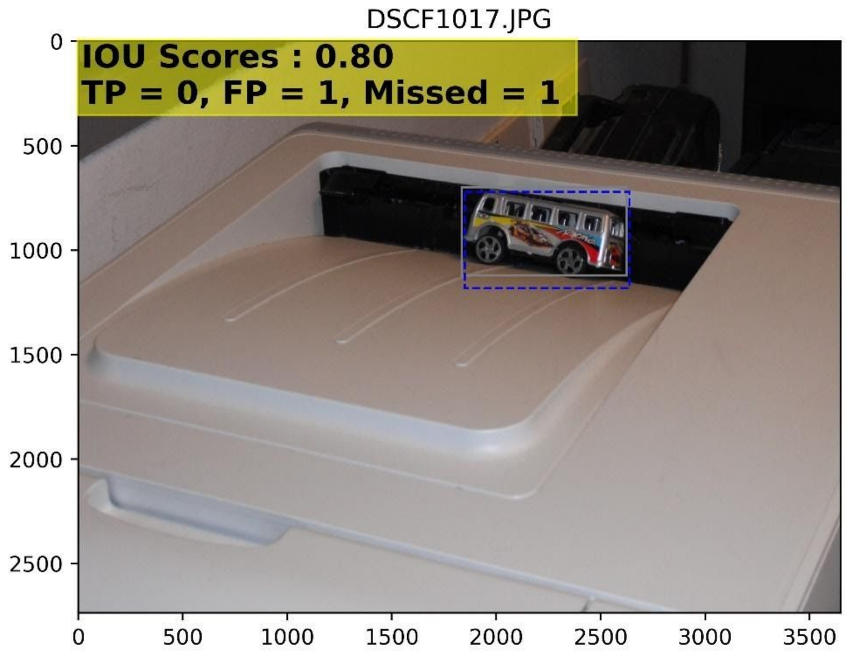
  If your submission exceeds the size limit in Moodle (500MB), send me the zip by email to: berkovitz1@mail.tau.ac.il

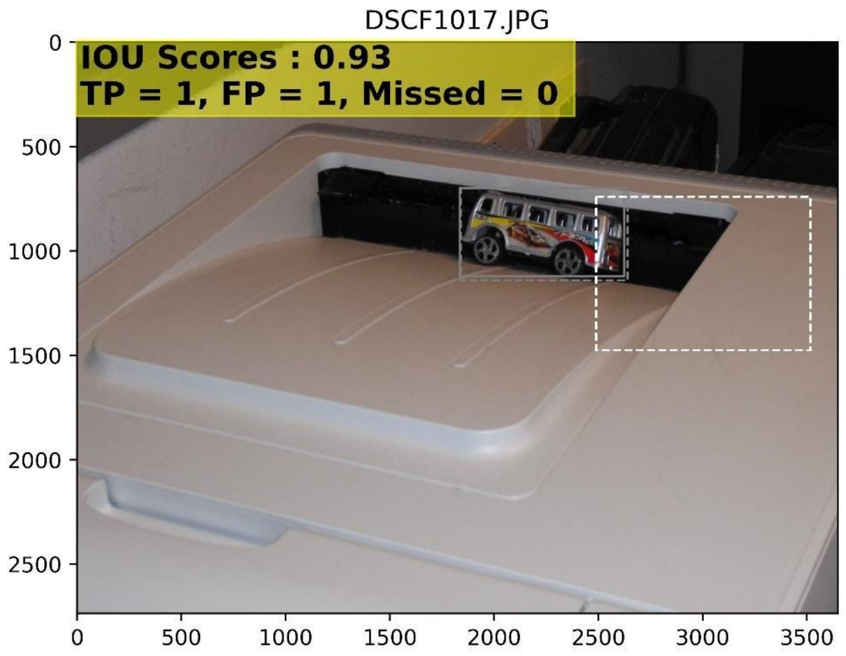
14. Project Exam (will take place the days following the submission date, specific dates will be announced later on):

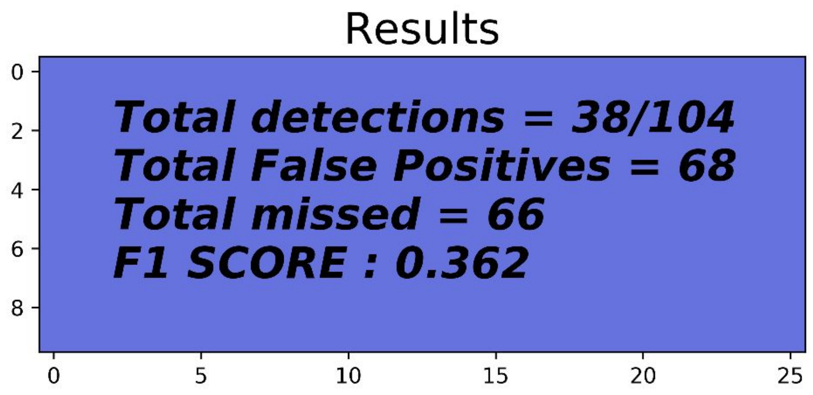
- Exam will be performed for each pair separately on Zoom.
- You will be asked to explain shortly about the algorithm you have implemented.
- Run the algorithm on the test-set and receive F1 Score.
- The test time is limited to 15 minutes!

15. Output examples of the test script (Main.py):








Results

Total detections = 38/104
Total False Positives = 68
Total missed = 66
F1 SCORE : 0.362