University at Buffalo
Department of Computer Science and Engineering
CSE 473/573 – Computer Vision and Image Processing

Spring 2020
TuTh 9:30AM - 10:50AM

Project #3 Submission Guidelines

1. **When to submit?**
   - Deadline: [5/13/20, 11:59PM + your remaining late day quota]
   - No submissions later than [5/16/20 11:59PM]

2. **Where to submit?**
   - Please submit to both **UBlearns AND AutoLab**.

3. **Additional REQUIREMENT**
   - For all projects your report should include a one sentence description of what each person did, and an approximate percentage of effort each person put in. This will NOT affect the project GRADE and will be informational only. If you all contributed equally, that is fine too.

4. **Submission package requirements**
   - **Face Detection in the Wild**

Submission Package Requirements:

Please submit your package to UBLearns in the format as shown below:

- **FaceDetection.zip**
  - **Model_Files** (a folder where you can put all your model related files, you can name your own directory)
  - **FaceDetection.py** (please use this name)
  - **results.json** (Please use this name, this is the result file generated by your script on the test case released on Piazza, under resource tab)
  - **Report.pdf** (please use this name)
  - **requirements.txt** (Optional, if your script requires special libraries other than numpy, opencv 3.4.2.17 and matplotlib, please include your library and corresponding versions here, each line shall only include one entry of library, in the format below:

    ```
    opencv-contrib-python=4.0.0
    opencv-python=4.0.0
    ```

    You can also refer to sample submission package for details.

)
  - **notes.txt** (Optional, leave any notes here if there is any special things you want TAs know regarding your submission)

Please submit **only [FaceDetection.py] script** to AutoLab.

It is your responsibility to make sure you've submitted exactly the same python script to both systems. Please submit only those required files.

Your script shall be in one python file only, the zip file shall be .zip format.

You need to download the test case from piazza (named as "test_images.zip" under resource tab, and generate results.json over this test set. You must run this ahead and include the results.json in your submission package.)

Json File Requirements:

A json results file is required to be submitted as mentioned above, please refer to section 3.3 of the project's handout regarding the format of the json file. You can also refer to the sample submission package for details.

Some students asked how to report results if there are multiple detected faces in one image, below is the format for this scenario:

results.json

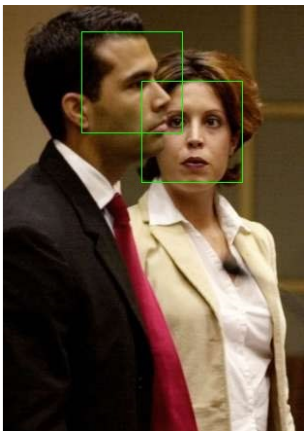
```
[...,
{"iname": "img_432.jpg", "bbox": [81, 30, 104, 104]},
{"iname": "img_432.jpg", "bbox": [143, 81, 104, 104]},
...]
```

In short, you can enter multiple entries if there are multiple faces in the same photo.

Note that there is a **json file checker** posted on piazza under resources as well. You **must** use this tool to check the format of your json file to see if it meets the requirement. This tool can also annotate your findings on the photos if you provide the images.

In order to run the script, you need to put your [results.json] under the ***json*** folder and then run the [json_checker_annotator.py] script. This script will then print out the results of the check.

If you also would like to overlay your detected boxes with the corresponding images, you can put all your tested images under ***imgs*** folder, the script will then help draw boxes over the photos based on your results.json. The annotated images will be saved to the [annotated] folder. Below is an example of the annotation.



Code Execution Requirements:

Your code will be executed with the following commands:

```
python FaceDetector.py [image-directory]
```

Please do not any use any flags to pass img-directory arguments like:

```
python FaceDetector.py ---img_directory [image-directory]
```

Below is an example of how to pass arguments without flags (skip this if you already know).

The contents of the demo script:

```
#  sysarg.py
import sys

print("Total lenth of sys.argv is {:3d}".format(len(sys.argv)))

for i, arg in enumerate(sys.argv):
    print("The {} th argument is {}".format(i, arg))
```

Run with following commands:

`python sysarg.py test1 test2`

The output:

```
Total lenth of sys.argv is   3
The 0 th argument is sysarg.py
The 1 th argument is test1
The 2 th argument is test2
```

Directory Processing Requirements:

You can skip this section if your script is written on Mac or Linux OS.

Please try not to hard code any backslash including “\” or “\\” in your directory process. This only works in windows but the code will be tested on Linux. You may use os.path.join(image_directory) to wrap your directory in order to make it cross os compatible.

- **Alternative Projects**
  - **i. Submission packaging**

    You package shall be named as one of the following depends on your choice:

    - Touch-lessFaceEnabledTime-clock.zip
    - VisualWelcomeCenter.zip
    - VirtualWall.zip
    - MobileRetriever.zip

    The package shall be zipped in .**zip format.**

    The contents to be included in the package:

    - Report
      - Cover page
      - 1-2 page description of the capabilities of your system
      - description on what you felt where the major challenges you overcame and what you would have improved if you had more time
    - src (all scripts in this folder, reference need to be marked with source)
    - Demo_video (a video “suitable for advertisement” that demonstrates your capabilities. Not something just talking to the professor. Just show your capabilities, with optional overlays describing capabilities.
    - Documentation
      - Software / API that needs to be installed/imported
      - Any reference (blogs, papers, github repo etc.)
      - If you have a driver to run the program, please include a description of how to run that as well.

  - ii. **Submission**

    Please submit the whole package to UBLeanrs.