

CASE STUDY

AI Product Developer

Context

Our software transforms automatically real-time video streams into mobility data. The output data is usually directly integrated into our client's daily operations software to better monitor and optimize their infrastructure (traffic lights, tolling systems, etc.). As those systems are very sensitive, our software needs to be very reliable and accurate.

Consequently, an important part of our daily job consists of auditing and maintaining our installed base to guarantee a very high level of availability and accuracy.

Your mission

As a new member of Wintics Product team, your goal is to design a solution that will help operations team to challenge the accuracy of our installed base.

To audit an object detection model, aside from the usual metrics, one can look at how good the detections are in certain zones of interest in a video. To do so, you will have to generate a tool that screenshots object detections in specific zones of videos for audit purposes. The code will have to meet our requirements of code standards.

Expectations

You'll have to implement a tool which can take screenshots when the objects {car, truck, motorcycle, bus} are detected inside the area of interest. The center of a bounding box is the reference point to determine the location of an object.


All detections inside the area of interest need to appear on the screenshot with a bounding box respecting the color convention you are given.



Those screenshots will have to respect the following name pattern:
"folder_path/screenshots/highway_{i}/{hour}h/{timestamp}.png"

With timestamps following ISO 8601 with timezone information (precision in seconds), *i* being the index of the point of view (1 or 2) and *hour* the beginning time of the video (9h or 18h).

Note: The exact day is not available, therefore you are free to choose one that suits you. Nevertheless, hours, minutes and seconds are available. Furthermore, we do not expect more than one screenshot per second.



Your code has to meet the following requirements:

- It needs to follow PEP8 standards. We recommend you to use pylint and flake8. (See <https://www.pylint.org/>, <https://flake8.pycqa.org/en/latest/>).
- The code must be modular. The main script has to be easy to read. Keep separate features in different files.
- Functions have to be tested through unit tests. You'll have a test folder "test" containing all your unit tests, the command "python -m unittest discover -s test" must run smoothly
- At least 3 tests are expected.
See <https://docs.python.org/3/library/unittest.html>.
- The code must be well documented.
- The main script may only accept as input argument a configuration file (.json or .yaml). No parameters must be written inside the code. e.g: video path, area of interest...

Note: it is not required to analyze the videos in their entirety (especially if you don't have a GPU), you can skip frames or start in the middle of videos.

Toolbox

To design and test your tool, you are provided with the following data:

- 4 videos: 2 different points of view (highways) * 2 videos each
- an area of interest for both points of views (a rectangle)
- color code for the detection visualization (e.g. which color for cars, etc.)

You can download the data following this link: https://case-screenshots.s3.eu-west-3.amazonaws.com/videos_param.zip

For detection, you'll have to use this fasterrcnn model provided by the library « torchvision »:
`model = torchvision.models.detection.fasterrcnn_resnet50_fpn(pretrained=True)`

At this stage of development, do not retrain this model as it is good enough for what is expected.

To convert the category id predicted by the detection model to the category name, see <https://github.com/nightrome/cocostuff/blob/master/labels.md>

Deliverables

Your work should be uploaded to a private git repo, including a readme to explain your project and how to run it. **Please send us the git link at the latest the evening prior your recruitment meeting.**

In between, feel free to add any functionality you see useful to the system, feel free to ask any questions.