

Junior ML test task

You have a fixed camera positioned to observe the area where an aircraft is readied for departure. Your task is to ensure that there are no vehicles on the stand before the aircraft arrives. If any vehicles are detected inside the stand boundary, you should notify the ground handler.

Requirements&common rules:

1. `>=python3.8`
2. The choice of language won't affect your evaluation, you can use either Russian or English.
3. Post your solution on your github and send us the link.
4. Deadline: 1 week after getting the assignment.

Task:

1. Input data:
 - a. (video) Folder with several short videos;
 - b. (annotation) Time intervals during which at least one vehicle is within the stand boundaries. The JSON file with keys `video_name`, and values list of pairs `(start_frame_number, end_frame_number)`. Where
 - i. `start_frame_number: int` — number of frame in video when any vehicle entered inside boundary
 - ii. `end_frame_number: int` — number of frame in video after which all vehicles left outside boundary

For example:

```
{"video_2.mp4": [[40, 43], [57, 60]]}
```

- c. (annotation) Stand boundaries in the JSON file with keys name of video and values polygons like `[[x1, y1], [x2, y2], ...]` where `x_i, y_i` — coordinate of pixel. Polygons have a minimum of 3 points and they are **not** always convex.

2. What we expect in return:

- a. Final results on test and train sets and their analysis, at a minimum:
 - i. What metrics do you use and why do you choose them? We expect from you here:
 - 1. List of metrics with explanation why you chose them.
 - 2. Text file with this metrics calculated on test set. Test set should be at least 20%, you should choose it by yourself.
 - 3. python script which calculate metrics on video + time intervals + polygons (we'll use it on unseen test)
 - ii. What are the limitations of your solution and how can you beat the remaining challenges?
- b. A brief explanation of how you selected technologies for your solution.
- c. Github repository with python code which process video. We evaluate the code for its readability and reusability.
- d. The script on python3 which is processing video file and save results into JSON file the same as the JSON file with time intervals. The script has following arguments:
 - i. `video_path` — str, path to video which we want to process
 - ii. `polygon_path` — str, path to JSON with boundaries for this video
 - iii. `output_path` — str, path to JSON file

For launching this script, we don't need to install anything special, ideally we clone your github repository and execute one `.sh` file.