# Programming challenge for Perception team member in Kopernikus

• What did you learn after looking on our dataset?

Dataset consist of total 1080 images captured through different cameras fit into different angles in the parking garage. As mentioned in task the format of filenames is different in some cases. The size of captured frames is also different in some cases. Images are captured in different lightning conditions as I can see some of them are captured in natural sunlight and some in darker conditions. Also, I can say that this type of dataset mostly creates a problem while we train machine learning algorithms for object recognition as most of images are similar and they induce high bias values. So, it is needed that the dataset should be pre-processed and similar images should be removed.

• How does you program work?

My program is simple program which can be improvised in much better way. I created the **task.py** file which contains the **delete_similar** function. This function required input as path to the dataset in which images are present. This function iterates through images and performs similarity check between two consecutive images i.e., between previous image and current image.

First, read all the images using **Imread** method of OpenCV and use **preprocess_image_change_detection** function from imaging_interview.py script for pre-processing. It converts the images into grayscale, applies gaussian blurring and mask out specific areas which is required for performing similarity check. After, pre-processing is done each consecutive pair of image i.e., previous image & current image is passed through **compare_frames_change_detection** function to calculate the score that represents the difference between those images. If this score is lesser than the threshold score (which already given as input parameter to **delete_similar** function) indicates images are similar and have very small difference between them. And finally, by using **os.remove** method the current image is removed from the dataset. I also comment out the code for moving similar images (current image) into different folder which can be used for future purpose.

• What values did you decide to use for input parameters and how did

you find these values?

Minimum Contour Area and Score threshold are two important parameters which are used as input for delete_similar function.

- **min_contour_area** - It determines whether or not the contour area of a certain location can contribute to the overall score of that particular image when compared to the preceding image. If this value is lower small differences are taken into the account while bigger values consider larger differences. After testing with various settings and test cases to see how they affected in comparison and removal of similar images. After multiple attempts I found that if it is set to 100 program produces good results.
- **score_threshold** – It decides the pair of images are similar or not. If calculated score for pair of image is smaller than score_threshold value it indicates that images are similar and if not then images are different. While experimenting with different values I found that 10000 is suitable in this case.

Both this values are found by running task.py script multiple times as trial and error method for getting better results overall.

• What you would suggest to implement to improve data collection of

unique cases in future?

It depends on which task we have to perform and for what purpose. In order to increase the data collection of unique cases in the future, we must guarantee that the dataset will contain a variety of images which are captured in different scenarios with different environmental conditions and lightning condition, Also capture the images with multiple objects in it. This will aid and expand coverage of uncommon instances. It also helps to object recognition algorithms perform better to identify multiple objects in multiple scenarios and conditions.

• Any other comments about your solution?

My solution is suitable for small cases with small datasets. There is lot of scope for improvement as currently my solution only taking consecutive images into account for similarity check, it can be improved. Also, I commented the code which can help to track the removed images and can be used for future purpose.