# Report first project “3D perception”.

1. INTRODUCTION

The aim of this project is to create an object detection network able to identify pedestrians in a real road environment. In order to achieve this goal, we used a YOLOv8n neural network, developed by Ultralytics. The network has been trained using images and labels coming from the KAIST dataset.

1. KAIST DATASET

It has been developed by a team of researchers of the Korea Advanced Institute of Science and Technology (KAIST). It comprehends 95k color-thermal pairs of images, their resolution is about 640x480, and every scene has been captured at 20 images per second. The thermal and the RGB cameras were mounted on the top of a prototype vehicle. All the image pairs were manually annotated classifying the pedestrians in the scenes in three different classes: person, people, and cyclist, these annotations are then reported into xml files, for a total of 103˙128 dense annotations and 1˙182 unique pedestrians.

The dataset comprehends scenes for both day and night scenarios; in our project we worked only with a subset of the night data (set 3 v0, set 3 v1, set 4 v0, set 4 v1, set 5 v0, set 9 v0, set 10 v0, set 11 v0) for a total of 18˙136 images and 7.3GB of storage.

1. YOLOv8

It is the latest version of the acclaimed real-time object detection and image segmentation model. YOLOv8n is built on cutting-edge advancements in deep learning and computer vision, offering unparalleled performance in terms of speed and accuracy. Its design makes it suitable for various applications and easily adaptable to different hardware platforms.

1. OUR WORK

3.1. DATA PREPARATION

The first step was to download the dataset from the following link: <https://soonminhwang.github.io/rgbt-ped-detection/>.

As we said before, the KAIST annotation format is xml, while the YOLO networks work with txt files. The problem of course is not only in the format but also in the content of the labels files which were substantially different.

The original annotations were very verbose and for every object identified they added many lines in the file. On the other hand, the YOLO format is much simpler and more direct, using only one line for every object in the label file.

We solved this problem by developing a parser code in python, which, starting from the xml annotation extracts the information for every object in terms of coordinates of the height, width, and the top-left vertex of the bounding boxes, in relative coordinates and the class number. These data were processed by the parser and translated in the correct YOLO format in the new txt file.

* 1. TRAINING

We tried different solution for training the NN:

* A training from scratch for 15 epochs but only on a smaller portion of the dataset comprehending only 7k images.
* For the second attempt we enlarged our dataset to over 18k images, and we imported a pretrained model on the COCO dataset created by the authors. The training on our custom dataset was carried on for 25 epochs.
* A third training made from scratch for 30 epochs, on the enlarged dataset.

We divided the training sessions in step of 10 or less epochs due to the time constraints imposed by the 24 hours limit for a single task in the HPC.

The results for the different solutions were quite different:

* For the first attempt the results were terrible with a maximum mean average precision (mAP) of 0.25.
* Thanks to the enlarged dataset and the pretrained model, this configuration led to the best results we obtained, with a mAP of around 0.97.
* The last case led to a mAp of 0.96.

1. CONCLUSIONS

In conclusion, we can say that the detection task is accomplished with satisfying results especially for the ‘person’ and ‘people’ classes, while for the ‘cyclist’ one the detection is not so accurate because of the lack of images.

Valerio Tiri, Federico Rovighi

20/12/2023