

# Concepts and Models of Knowledge Engineering

## REPORT

Domain : Smart City

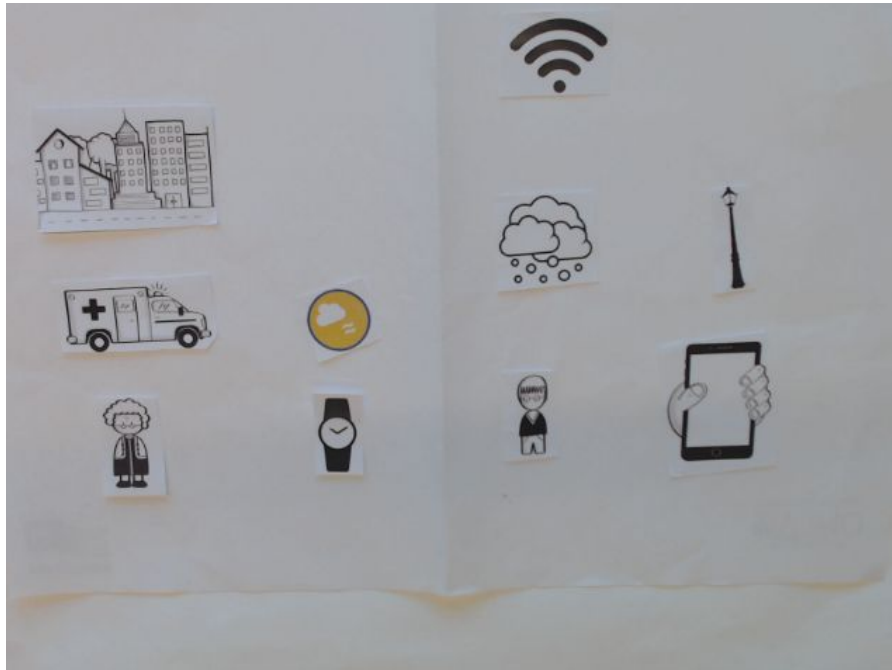
<b>Group number:</b>	04
----------------------	----

Group member 1	
<b>Name:</b>	Sofiia Badera
<b>Student ID:</b>	11715248
<b>E-mail address:</b>	a11715248@unet.univie.ac.at

Group member 2	
<b>Name:</b>	Aras Saeidian
<b>Student ID:</b>	11821135
<b>E-mail address:</b>	a11821135@unet.univie.ac.at

# Introduction

For this project we used Smarty City Domain to train the network and recognize the objects. There are 6 photos according to the Smart City domain and extra 4 photos for testing the training process. Car (from hospital domain), Watch (from production domain) , Cloud (from ski tourism domain) and Woman (from university domain) are extra objects.



10 classes were used: boy, woman, watch, scanner, phone, lamp, internet, car, cloud, street.

Because of the lack of resources at home, we used OMiLab resources like Camera, Computer, GPU and softwares we needed.

## Net 1

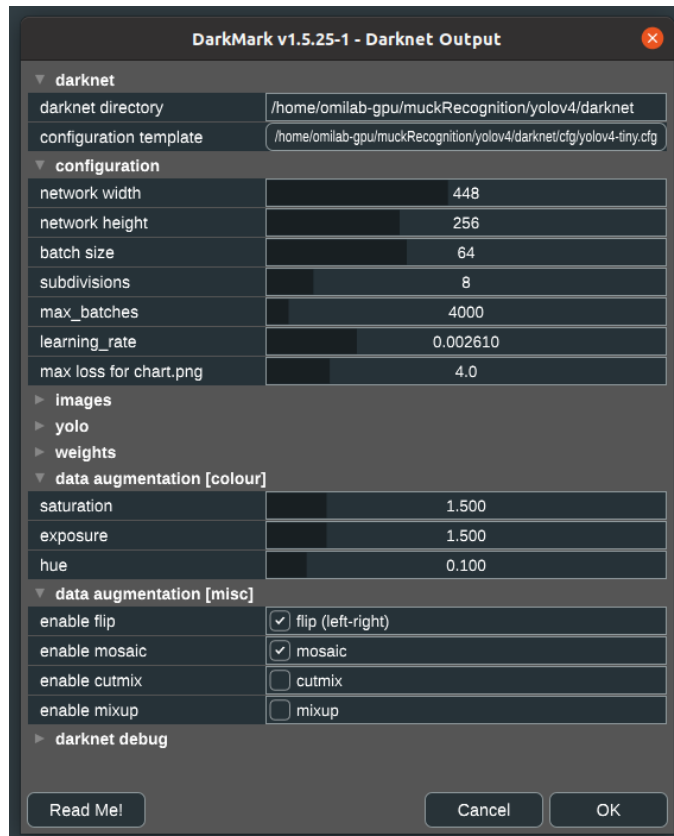
### Configuration for Net 1:

Batch size: 64

Subdivisions: 8

Max bathes: 4000

Other configurations are shown in the figure below, according to software DarkMark that was used:



We also used augmented figures produced by DarkMark with different techniques like flipped, noise and blur, perspective, affine.

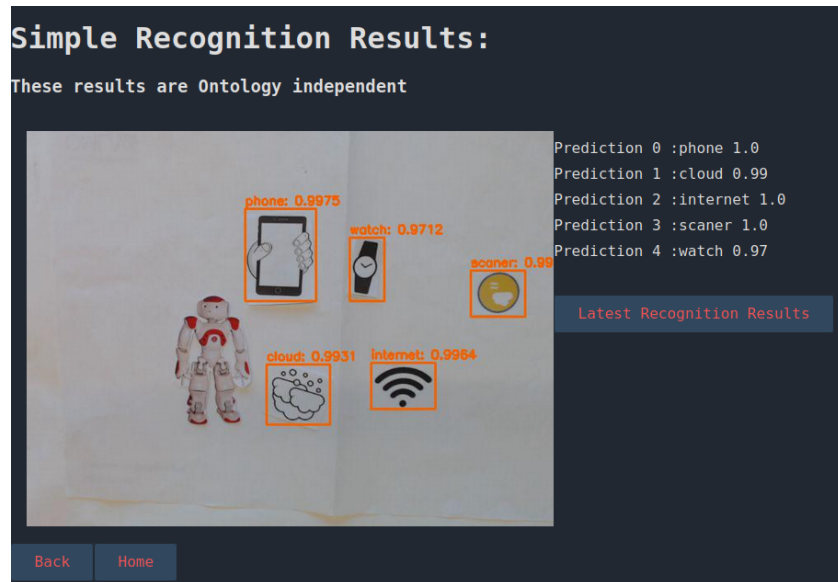
The training took around 14 minutes as we see in the figure below:

```

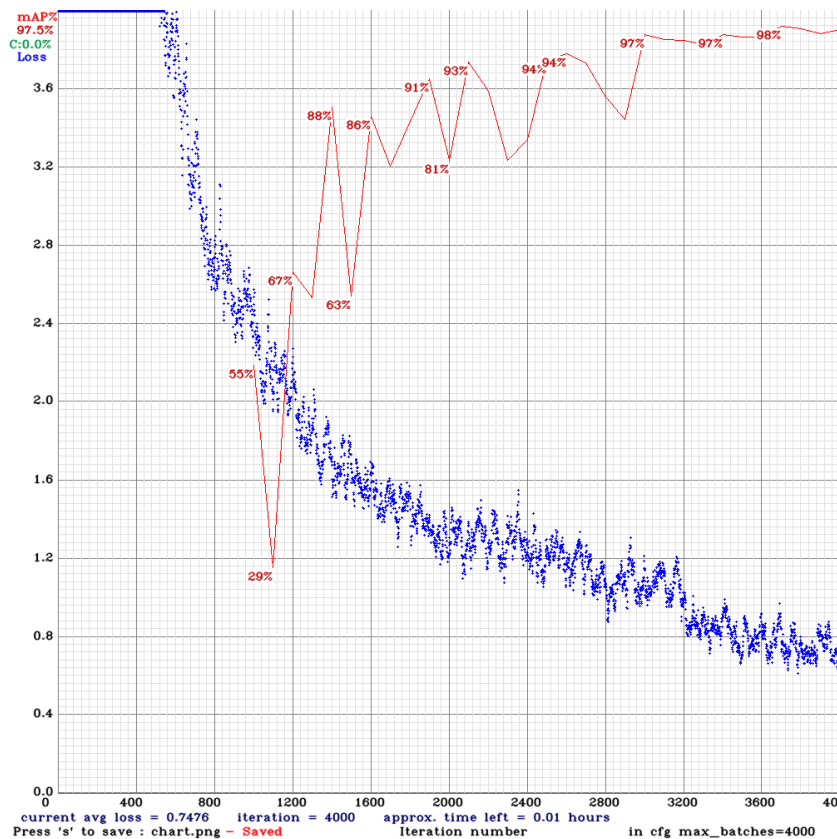
omilab-gpu@omilabgpu: ~/Desktop/CMKE/WS22/G4/N1
.data /home/omilab-gpu/Desktop/CMKE/WS22/G4/N1/N1.cfg"
User time (seconds): 4013.77
System time (seconds): 1048.07
Percent of CPU this job got: 602%
Elapsed (wall clock) time (h:mm:ss or m:ss): 13:59.88
Average shared text size (kbytes): 0
Average unshared data size (kbytes): 0
Average stack size (kbytes): 0
Average total size (kbytes): 0
Maximum resident set size (kbytes): 1559688
Average resident set size (kbytes): 0
Major (requiring I/O) page faults: 0
Minor (reclaiming a frame) page faults: 1451051
Voluntary context switches: 23832364
Involuntary context switches: 110438270
Swaps: 0
File system inputs: 0
File system outputs: 2504472
Socket messages sent: 0
Socket messages received: 0
Signals delivered: 0
Page size (bytes): 4096
Exit status: 0
(base) omilab-gpu@omilabgpu:~/Desktop/CMKE/WS22/G4/N1$

```

After training we used OMiLab camera and ran the server on Computer of Lab to test the network.



We also decided to use a robot figure that was in the Lab to test the network with an unusual shape and complex object. For recognized objects we got acceptable results with this configuration and time, although sum objects were flipped but prediction rates were high.



As it has been shown in the chart, Mean average of precision equals 97.5% after 4000 iterations but the average loss is 0.

## Net 2

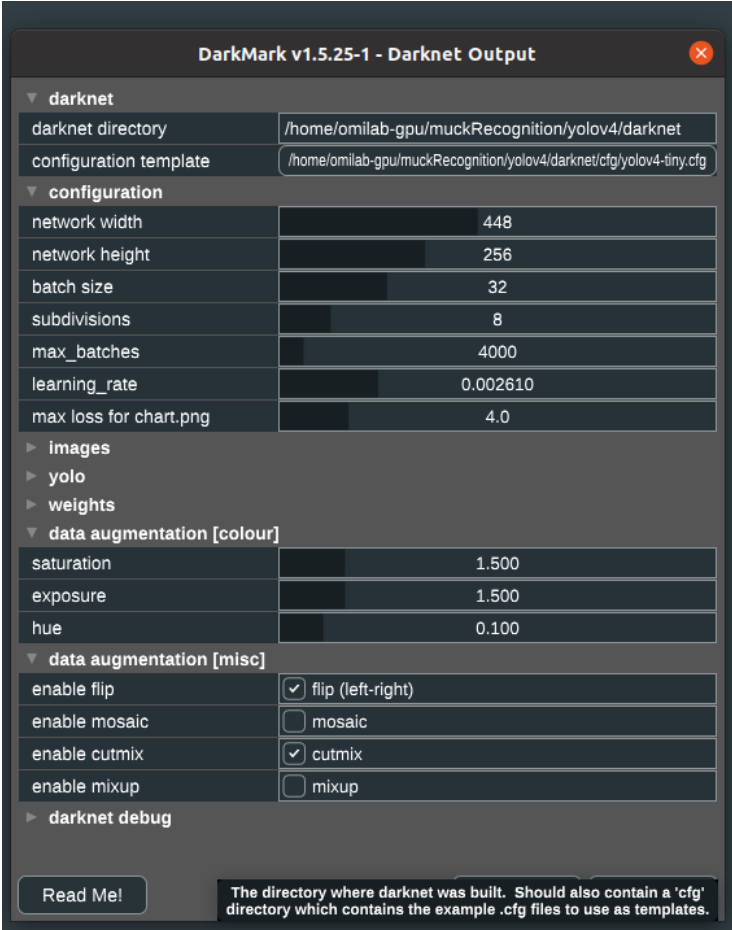
### Configuration for Net 2:

Batch size: 32

Subdivisions: 8

Max bathes: 4000

Other configurations are shown in the figure below, according to software DarkMark that was used:



**DarkMark v1.5.25-1 - Darknet Output**

- darknet**
  - darknet directory: /home/omilab-gpu/muckRecognition/yolov4/darknet
  - configuration template: /home/omilab-gpu/muckRecognition/yolov4/darknet/cfg/yolov4-tiny.cfg
- configuration**

network width	448
network height	256
batch size	32
subdivisions	8
max_batches	4000
learning_rate	0.002610
max loss for chart.png	4.0
- images**
- yolo**
- weights**
- data augmentation [colour]**

saturation	1.500
exposure	1.500
hue	0.100
- data augmentation [misc]**

enable flip	<input checked="" type="checkbox"/> flip (left-right)
enable mosaic	<input type="checkbox"/> mosaic
enable cutmix	<input checked="" type="checkbox"/> cutmix
enable mixup	<input type="checkbox"/> mixup
- darknet debug**

[Read Me!](#) The directory where darknet was built. Should also contain a 'cfg' directory which contains the example .cfg files to use as templates.

We also used augmented figures produced by DarkMark with different techniques like cutout, flipped, noise and blur, perspective.

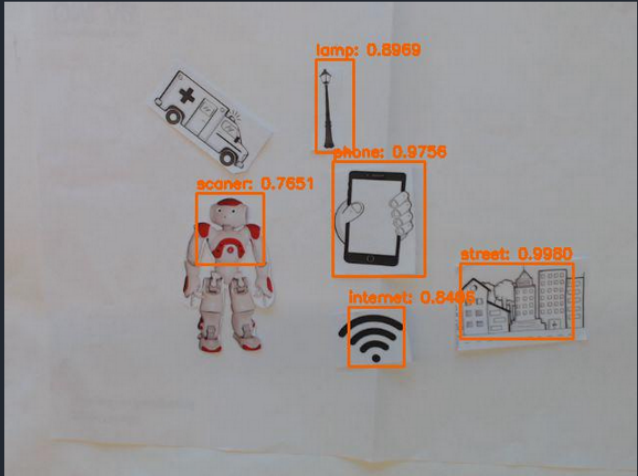
As we expected, because of less batch size in compare with Net 1, training lasted faster and took about 7.28 mins as we see in the figure below:

```
omilab-gpu@omilabgpu: ~/Desktop/CMKE/WS22/G4/N2
.data /home/omilab-gpu/Desktop/CMKE/WS22/G4/N2/N2.cfg"
User time (seconds): 986.58
System time (seconds): 223.72
Percent of CPU this job got: 269%
Elapsed (wall clock) time (h:mm:ss or m:ss): 7:28.49
Average shared text size (kbytes): 0
Average unshared data size (kbytes): 0
Average stack size (kbytes): 0
Average total size (kbytes): 0
Maximum resident set size (kbytes): 1422216
Average resident set size (kbytes): 0
Major (requiring I/O) page faults: 0
Minor (reclaiming a frame) page faults: 351415
Voluntary context switches: 4533806
Involuntary context switches: 24603742
Swaps: 0
File system inputs: 0
File system outputs: 2364664
Socket messages sent: 0
Socket messages received: 0
Signals delivered: 0
Page size (bytes): 4096
Exit status: 0
(base) omilab-gpu@omilabgpu:~/Desktop/CMKE/WS22/G4/N2$
```

After training we used OMiLab camera and ran the server on Computer of Lab to test the network.

## Simple Recognition Results:

These results are Ontology independent



Prediction 0 :phone 0.98  
Prediction 1 :street 1.0  
Prediction 2 :scanner 0.77  
Prediction 3 :lamp 0.9  
Prediction 4 :internet 0.84

Latest Recognition Results

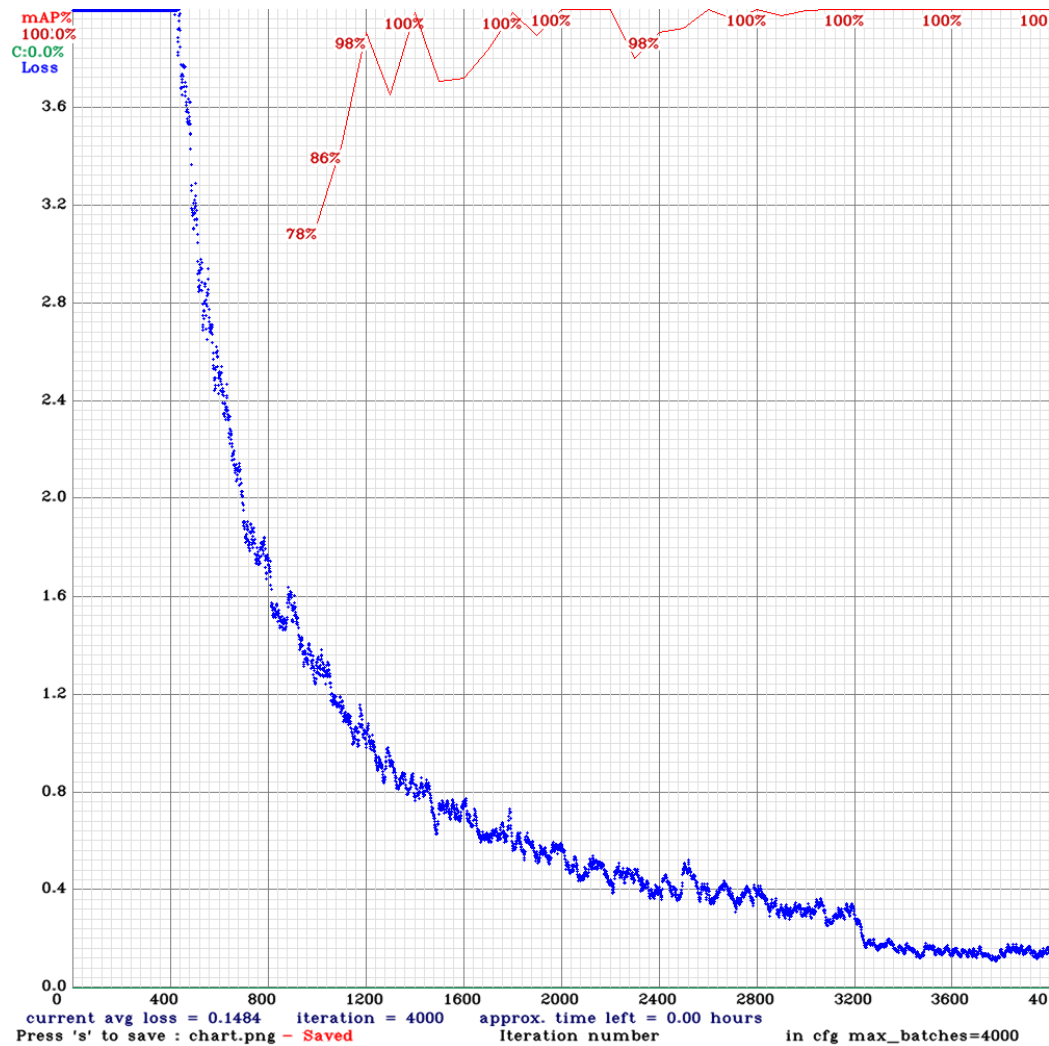
Back Home

Similar like in Net 1, we used a robot schema that was in the Lab to test the network with an unusual shape and complex object. For recognized objects we got less accuracy in comparison

with Net 1, as we see in prediction rates. Interesting that it recognized the robot as a scanner that we used for sensor medium:



It seems this result is because of the different color and also the circle shape of the head of the robot and its curly form.



As it has been shown in the chart, Mean average of precision is 100% after 4000 iterations but the average loss is 0.

## Net 3

### Configuration for Net 3:

Batch size: 128

Subdivisions: 8

Max bathes: 4000

Other configurations are shown in the figure below, according to software DarkMark that was used:



The screenshot shows the 'DarkMark v1.5.25-1 - Darknet Output' window. It contains several sections for configuring a darknet model. The 'darknet' section has fields for 'darknet directory' and 'configuration template'. The 'configuration' section has fields for 'network width', 'network height', 'batch size', 'subdivisions', 'max\_batches', 'learning\_rate', and 'max loss for chart.png'. The 'images' section has checkboxes for 'do not resize images', 'resize images', 'tile images', 'crop & zoom images', 'limit negative samples', and 'train with all images', along with a 'training images %' field. At the bottom, there are buttons for 'Read Me!', 'Cancel', and 'OK', and a status bar indicating the darknet configuration file to use as a template for this project.

DarkMark v1.5.25-1 - Darknet Output	
▼ darknet	
darknet directory	/home/omilab-gpu/muckRecognition/yolov4/darknet
configuration template	/home/omilab-gpu/muckRecognition/yolov4/darknet/cfg/yolov4-tiny.cfg
▼ configuration	
network width	448
network height	256
batch size	128
subdivisions	8
max_batches	4000
learning_rate	0.002610
max loss for chart.png	4.0
▼ images	
do not resize images	<input type="checkbox"/> do not resize images
resize images	<input type="checkbox"/> resize images to match the network dimensions
tile images	<input checked="" type="checkbox"/> tile images to match the network dimensions
crop & zoom images	<input checked="" type="checkbox"/> random crop and zoom images
limit negative samples	<input checked="" type="checkbox"/> limit negative samples
train with all images	<input checked="" type="checkbox"/> train with all images
training images %	80
▶ yolo	
▶ weights	
▶ data augmentation [colour]	
▶ data augmentation [misc]	
▶ darknet debug	
Read Me! Cancel OK	
The darknet configuration file to use as a template for this project.	

We decided to make more changes in Net 3 in comparison with Net 2 and Net 1. Different augmented techniques were used, such as random crop and zoom images. For more accuracy we chose 128 for batch size and expected more time for training.

Training took about 26 mins as we see in the figure below:

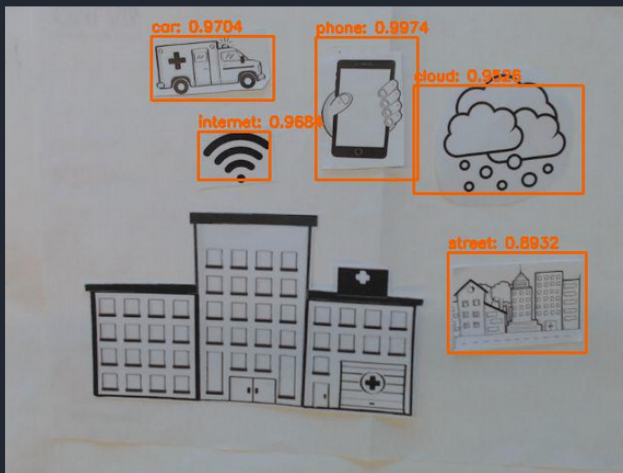


```
omilab-gpu@omilabgpu: ~/Desktop/CMKE/WS22/G4/N3
.data /home/omilab-gpu/Desktop/CMKE/WS22/G4/N3/N3.cfg"
User time (seconds): 4793.76
System time (seconds): 1188.33
Percent of CPU this job got: 381%
Elapsed (wall clock) time (h:mm:ss or m:ss): 26:09.95
Average shared text size (kbytes): 0
Average unshared data size (kbytes): 0
Average stack size (kbytes): 0
Average total size (kbytes): 0
Maximum resident set size (kbytes): 1719064
Average resident set size (kbytes): 0
Major (requiring I/O) page faults: 0
Minor (reclaiming a frame) page faults: 3628780
Voluntary context switches: 24093979
Involuntary context switches: 117290813
Swaps: 0
File system inputs: 0
File system outputs: 2318864
Socket messages sent: 0
Socket messages received: 0
Signals delivered: 0
Page size (bytes): 4096
Exit status: 0
(base) omilab-gpu@omilabgpu:~/Desktop/CMKE/WS22/G4/N3$
```

After training we used OMiLab camera and ran the server on Computer of Lab to test the network.

## Simple Recognition Results:

These results are Ontology independent



Prediction 0 :cloud 0.95 (recognised symbols: 0 00 0 0)  
Prediction 1 :phone 1.0  
Prediction 2 :street 0.89  
Prediction 3 :car 0.97  
Prediction 4 :internet 0.97

Latest Recognition Results

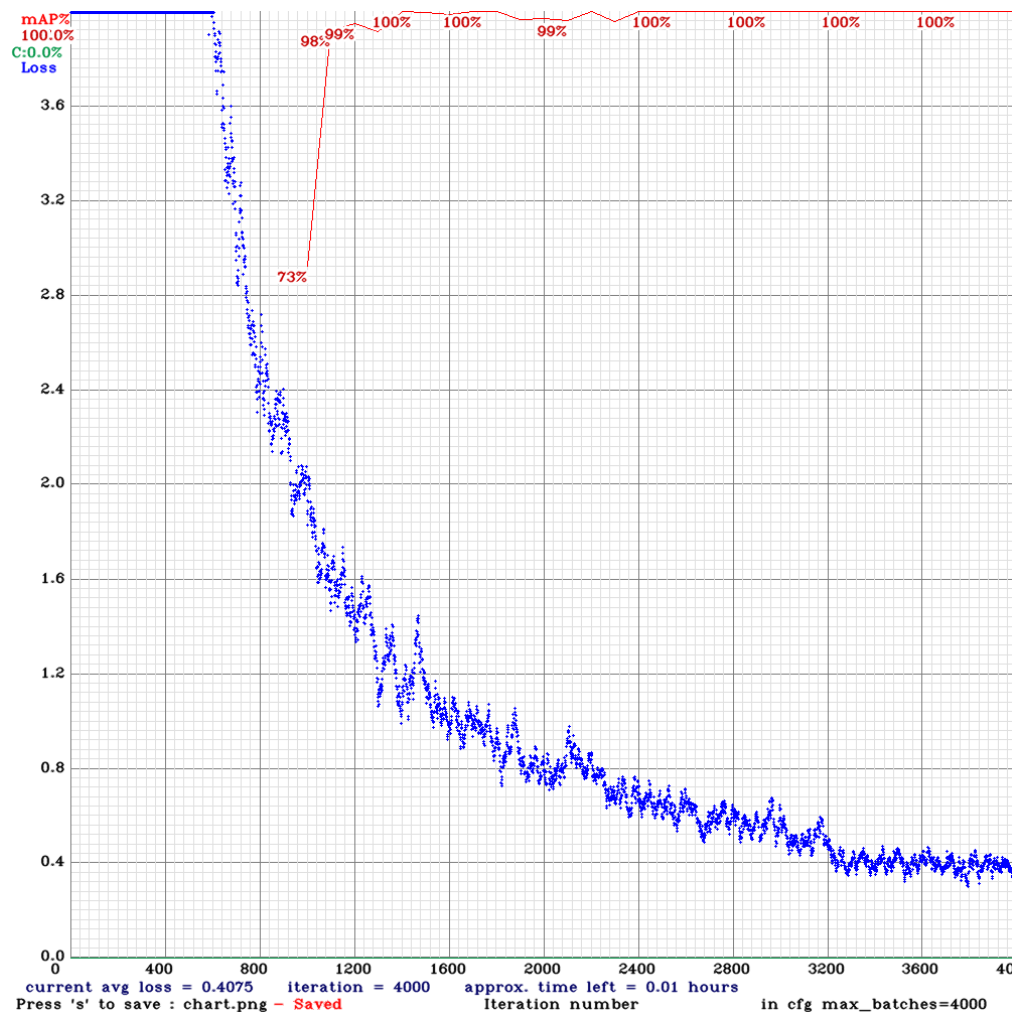
Back

Home

As it is seen in the result, in Net3, the car object could be recognized, but other objects have less prediction rates in comparison to Net 1, as results depend on different parameters. We used different augmented techniques in Net 1, Net 2 and Net 3.

On the other side, as it is shown in the chart below, in Net 3, we reached 100% mAP after 1200 iterations, but in Net 1, we had less stability to reach final mAP and it was a complicated path to reach 97.5%.

As we see, these 3 scenarios that have different parameters, give us different result with different accuracy.



As it has been shown on the chart, Mean average of precision is 100% after 4000 iterations but the average loss is 0.