GPI Open User AI

Version 2.0

GPI KOREA Co., Ltd.

http://www.gpikorea.com

Table of Contents

1. Introduction of GPI Open User AI

This manual explains how to create an ONNX file by learning with a user's dataset.

1.1. Prepare PC

Ubuntu 20.04 is recommended.

1.2. Creating a Python virtual environment using virtualenv

1. Install python3-virtualenv using apt (advanced packaging tool)

```
$sudo apt install python3-virtualenv
```

2. Create my-venv (Python virtual environment name) using virtualenv.

```
$virtualenv my-venv
```

3. Activate my-venv.

```
$source my-venv/bin/activate
```

4. When my-venv is activated, it will display as below.

```
(my-venv) training@training-WS-C246M-PRO-Series:~$
```

5. Download "trainer_ssdlite_sy.tara".
https://www.dropbox.com/scl/fo/1z1xzlu8a1j5xcz6llo0l/h?rlkey=lnxtq138y6fxte8t47lgq2a0v&dl=0

1.3. MobilenetV2 + SSDLite (Detector) training environment setting

1. Move the trainer_ssdlite_sy.tar file required for deep learning training to the prepared Ubuntu pc.

2. Use the command below to unpack trainer_ssdlite_sy.tar.

```
result:
(my-venv) training@training-WS-C246M-PRO-Series:~$ ll trainer
total 20
drwxrwxr-x  5 training training 4096 Sep 21 09:51 ./
drwxr-xr-x 22 training training 4096 Jan 30 11:32 ../
drwxrwxr-x 10 training training 4096 Sep 29 09:38 mobilenetV2_SSDliteV1/
drwxrwxr-x 10 training training 4096 Sep 29 09:38 mobilenetV2_SSDliteV2/
drwxrwxr-x 11 training training 4096 Feb  2 08:28 mobilenetV2_SSDliteV2_full_rfb/
```

3. Move to the trainer directory by using the below command.

```
$cd trainer
```

4. Select the model you want to use for training and move to its directory.
   (mobilenetV2_SSDliteV2_full_rfb in this manual)

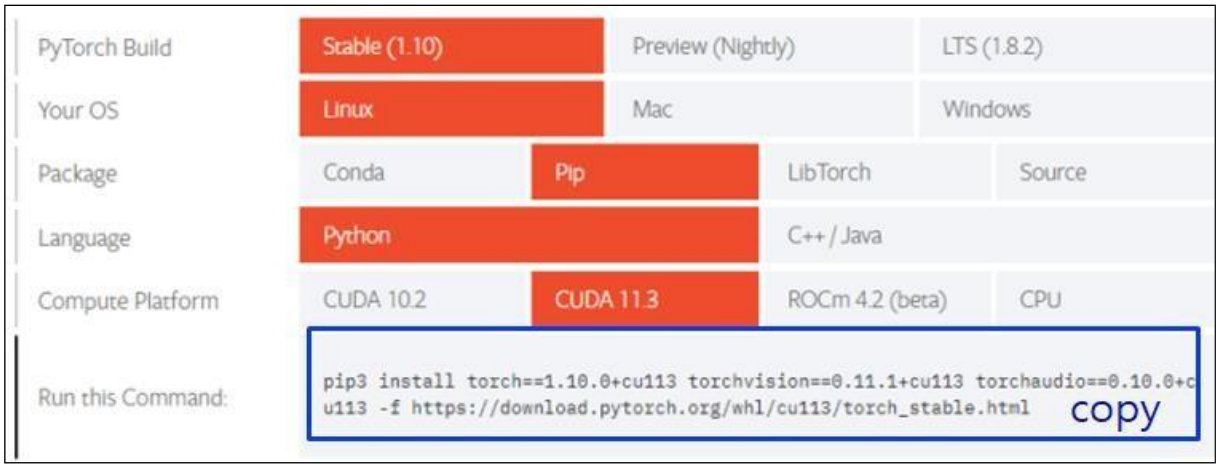
```
$cd mobilenetV2_SSDliteV2_full_rfb
```

	DDR usage	Storage space usage	Inference speed	mAP(PASCAL VOC 2007)
mobilenetV2_SSDLiteV1	2.9MB	4.5MB	16.6ms	0.69
mobilenetV2_SSDLiteV2	3.6MB	5.7MB	20.8ms	0.74
mobilenetV2_SSDLiteV2_full	3.8MB	7.4MB	23.5ms	0.745

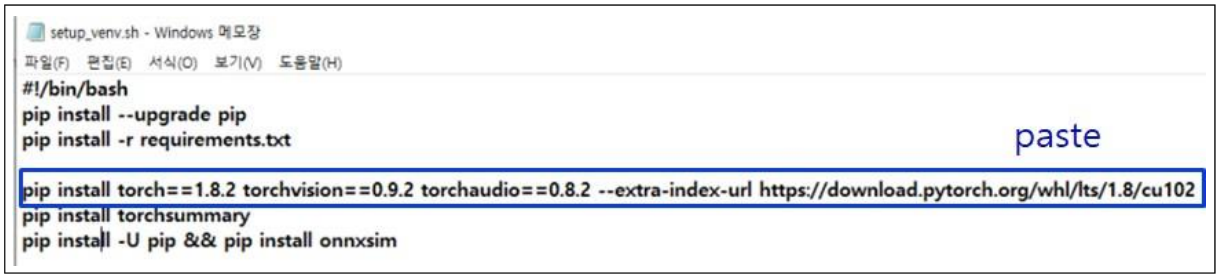
<Table 1 Memory, storage usage, inference speed, and mAP comparison table by model >

5. Edit the setup_venv.sh file.

Access https://pytorch.org/get-started/locally page then select torch and torchvision that match your CUDA version, and copy the completed command.
(PyTorch and TorchVision versions may differ from the image below.)



Replace the PyTorch installation command part in setup_venv.sh with the command copied above.



6. Run setup_venv.sh in the created virtual environment.

```
$./setup_venv.sh
```

7. You can check the packages installed by the setup_venv.sh script with the pip list command.

```
(my-venv) training@training-WS-C246M-PRO-
Series:~/trainer/mobilenetV2_SSDliteV2_full_rfb$ pip list
Package                 Version
------------------------ ----------
absl-py                 1.4.0
adamp                   0.3.0
cachetools              4.2.4
```

certifi	2022.6.15
…	
tensorboardX	2.5.1
torch	1.8.2+cu102
torchaudio	0.8.2
torchsummary	1.5.1
torchvision	0.9.2+cu102

1.4. MobilenetV2 + SSDLite (Detector) training

1. Prepare a Dataset

Download the VOC 2007 and 2012 datasets from the link below.

https://drive.google.com/file/d/12RmxmsmwN-u-UyWf2bLkEBRTyoCy6-IT/view?usp=sharing

2. Unzip the downloaded dataset.

```
$cd /home/training/trainer/mobilenetV2_SSDliteV2_full_rfb
$unzip PASCAL_VOC_Dataset.zip
```

result :

```
(my-venv) training@training-WS-C246M-PRO-
Series:~/trainer/mobilenetV2_SSDliteV2_full_rfb$ ll PASCAL_VOC_Datasettotal 16
drwxrwxr-x  4 training training 4096 Feb  2 08:29 ./
drwxrwxr-x 11 training training 4096 Feb  2 08:28 ../
drwxrwxr-x  5 training training 4096 May  3  2021 PASCAL_VOC_2007/
drwxrwxr-x  4 training training 4096 May 17  2021 PASCAL_VOC_2012/
```

3. When learning, class id must start from 1. You can designate 0 as unknown and start with 1 from a meaningful class.

4. Training

```
$python3 detector_train.py --dataset_root ./PASCAL_VOC_Dataset/PASCAL_VOC_2012 -
```

```
-dataset_domains train,validation --epochs 200 --class_path ./PASCAL_VOC.txt
```

```
result :
(my-venv) training@training-WS-C246M-PRO-
Series:~/trainer/mobilenetV2_SSDliteV2_full_rfb$ python3 detector_train.py --
dataset_root ./PASCAL_VOC_Dataset/PASCAL_VOC_2012 --dataset_domains
train,validation --epochs 200 --class_path ./PASCAL_VOC.txt
Start Training
Model: ssdlite320
Epochs : 200
Batch Size: 32
Learning Rate: 0.01000
Optimizer : AdamW
Scheduler: cosine

Downloading: "https://download.pytorch.org/models/mobilenet_v2-b0353104.pth" to
/home/training/.cache/torch/hub/checkpoints/mobilenet_v2-b0353104.pth
100.0%
2023-02-02 11:34:34.670912   epoch#0 start
[32 / 11540] - loss=14.548
```

5. Check the trained model files (model files are saved in the ./checkpoints directory every 5 epochs).

```
$ll ./checkpoints
```

```
result :
(my-venv) training@training-WS-C246M-PRO-
Series:~/trainer/mobilenetV2_SSDliteV2_full_rfb $ ll checkpoints/
total 88545760
drwxrwxr-x   2 training training     49152 Jan 26 11:01 ./
drwxrwxr-x 13 training training       4096 Jan 19 15:00 ../
```

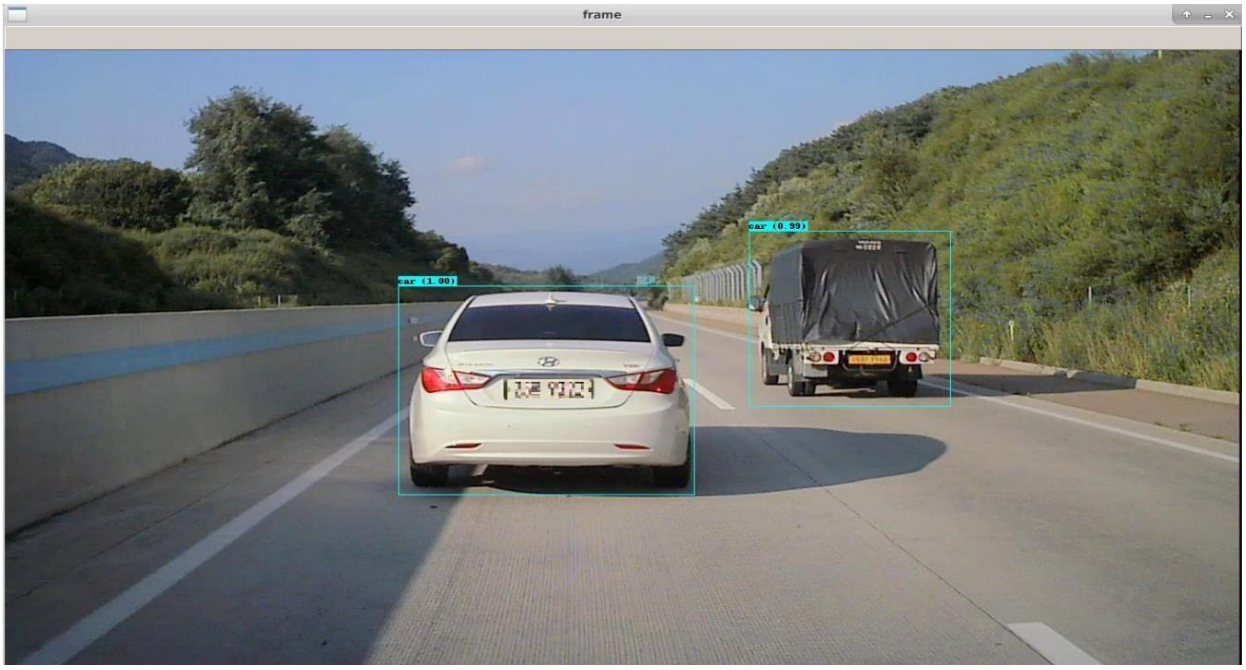
```
-rw-rw-r--  1 training training 92759334 Dec 23 09:50 ssdlite320_obj_10.pth
-rw-rw-r--  1 training training 92759334 Dec 23 10:48 ssdlite320_obj_15.pth
-rw-rw-r--  1 training training 92759334 Dec 23 11:46 ssdlite320_obj_20.pth
-rw-rw-r--  1 training training 92759334 Dec 23 12:45 ssdlite320_obj_25.pth
```

1.5. Convert the trained model to an onnx file

1. Use the command below to check the inference result and convert it to an .onnx file..

```
$python3 detector_single.py --class_path ./PASCAL_VOC.txt --weight
checkpoints/ssdlite320_obj_latest.pth --export { Image path you want to infer }
```

result :




```
(my-venv) training@training-WS-C246M-PRO-
Series:~/trainer/mobilenetV2_SSDliteV2_full_rfb $ ll ssdlite*
total 88545760
drwxrwxr-x  2 training training     49152 Jan 26 11:01 ./
drwxrwxr-x 13 training training      4096 Jan 19 15:00 ../
-rw-rw-r--  1 training training 92759334 Dec 23 09:50 ssdlite320.onnx
-rw-rw-r--  1 training training 92759334 Dec 23 10:48 ssdlite320.anchor
```

1.6. Validating the Trained Model

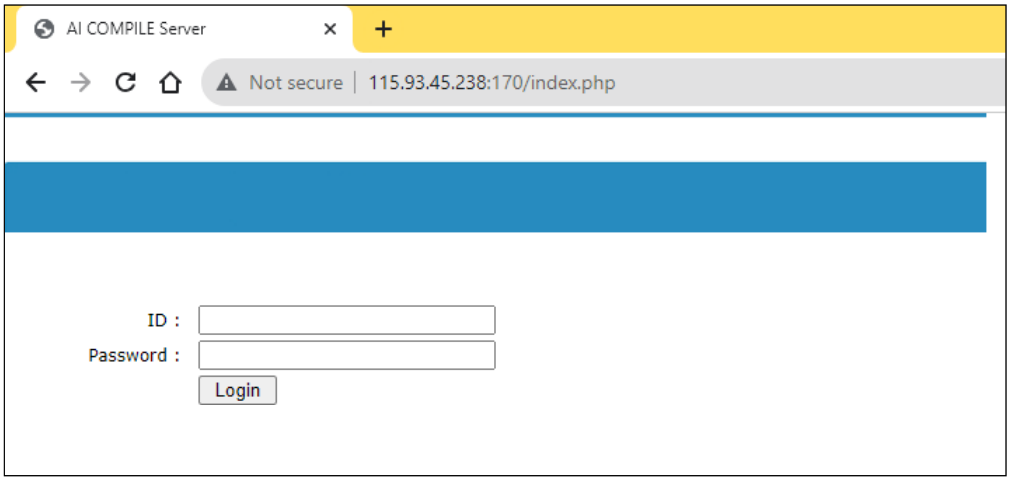
1. You can verify the trained model and check the mean average precision (mAP) using the command below.

```
$python3 detector_eval.py --dataset_root ./PASCAL_VOC_Dataset/PASCAL_VOC_2007/ --dataset_domains test --class_path ./PASCAL_VOC.txt --weight checkpoints/ssdlite320_obj_latest.pth
```

```
result :
AP(aeroplane) = 0.361
AP(bicycle) = 0.202
AP(bird) = 0.117
AP(boat) = 0.022
AP(bottle) = 0.000
AP(bus) = 0.238
AP(car) = 0.406
AP(cat) = 0.276
AP(chair) = 0.029
AP(cow) = 0.172
AP(diningtable) = 0.113
AP(dog) = 0.254
AP(horse) = 0.265
AP(motorbike) = 0.174
AP(person) = 0.447
AP(pottedplant) = 0.004
AP(sheep) = 0.070
AP(sofa) = 0.132
AP(train) = 0.162
AP(tvmonitor) = 0.085
mAP = 0.177
```

1.7. Convert the trained model to an onnx file

1. Prepare 4 files: the .onnx and .anchor files created above, the label.txt file used for training, and sample_dataset.zip (select and compress about 100 of the datasets used for training).
2. Connect to http://115.93.45.238:170
3. A web page as per below will appear.

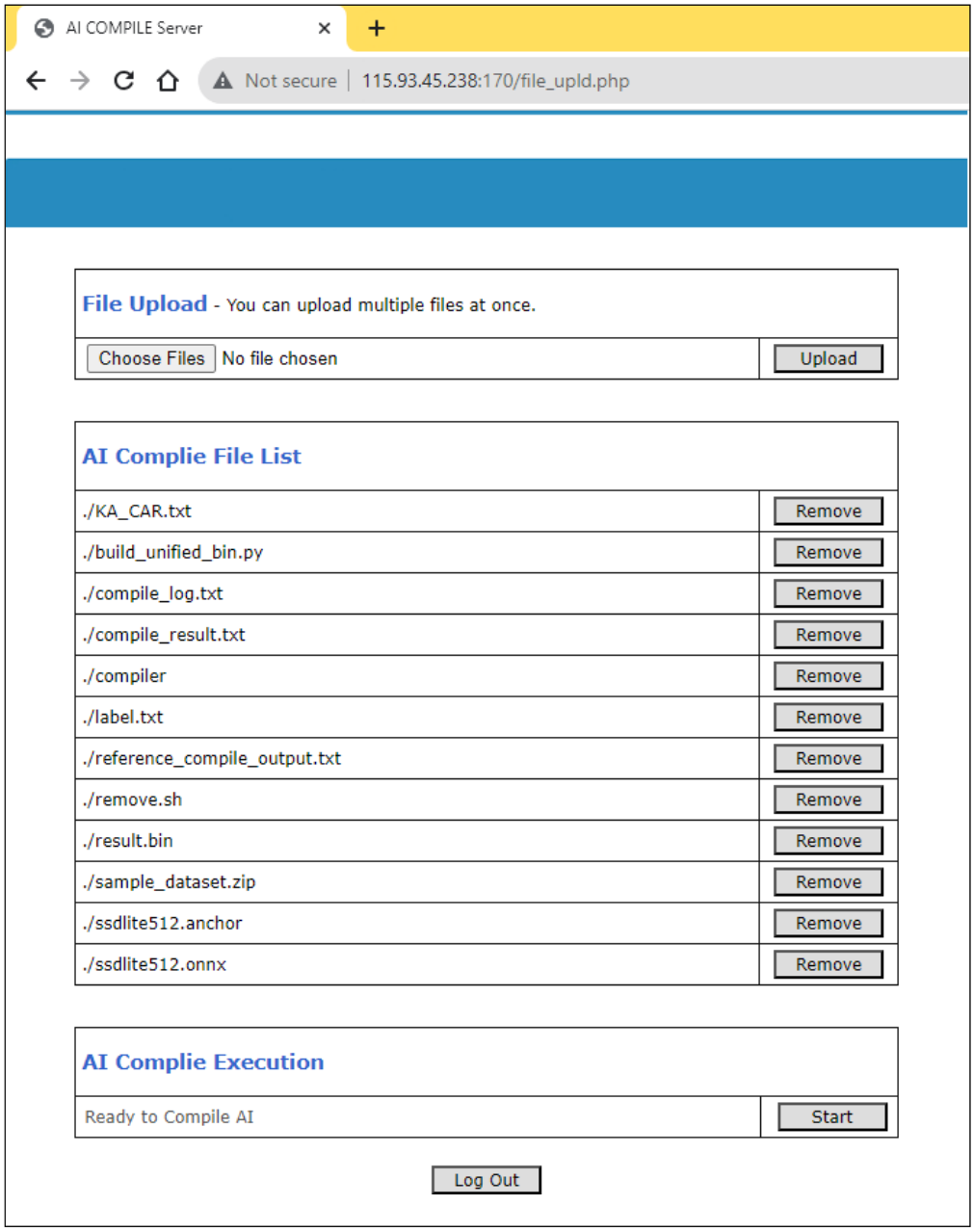


Log in with the login information below.

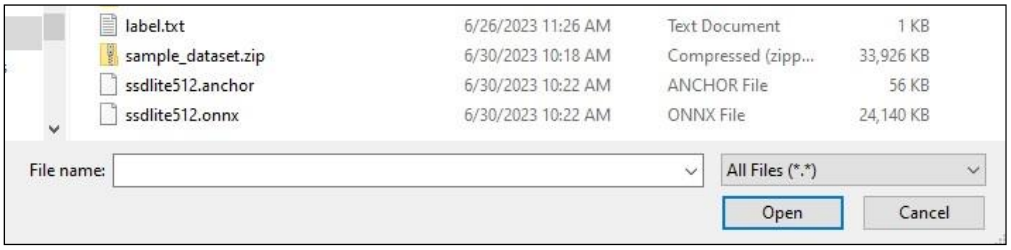
id	manager
pw	manager

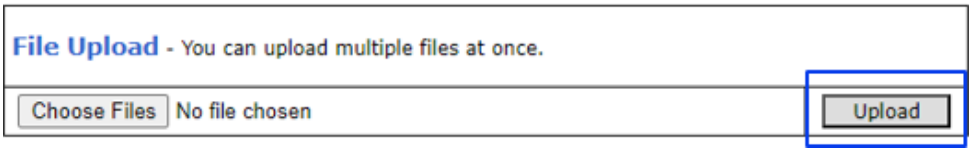
The start button is disabled because the files required for compilation are not uploaded.

4. A web page as per below will appear.

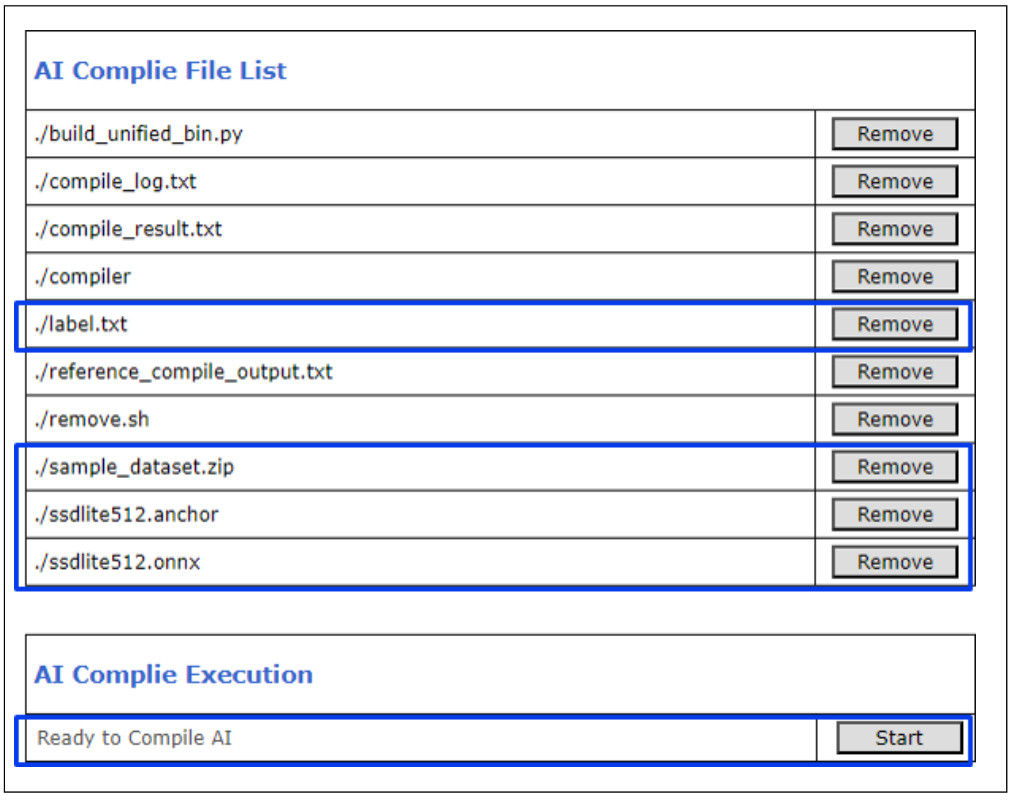


5. Click Select File in the File Upload section and upload *.onnx, *.anchor, sample_dataset.zip, label.txt files.

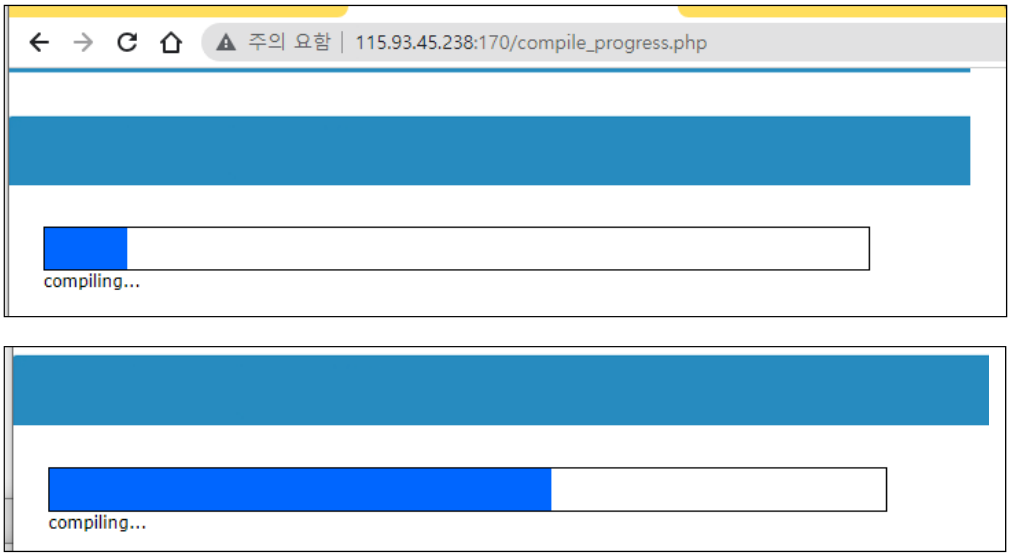




6. When the upload is finished, you can see that the start button is activated along with the list of uploaded files as shown below. Click the start button to start compiling.

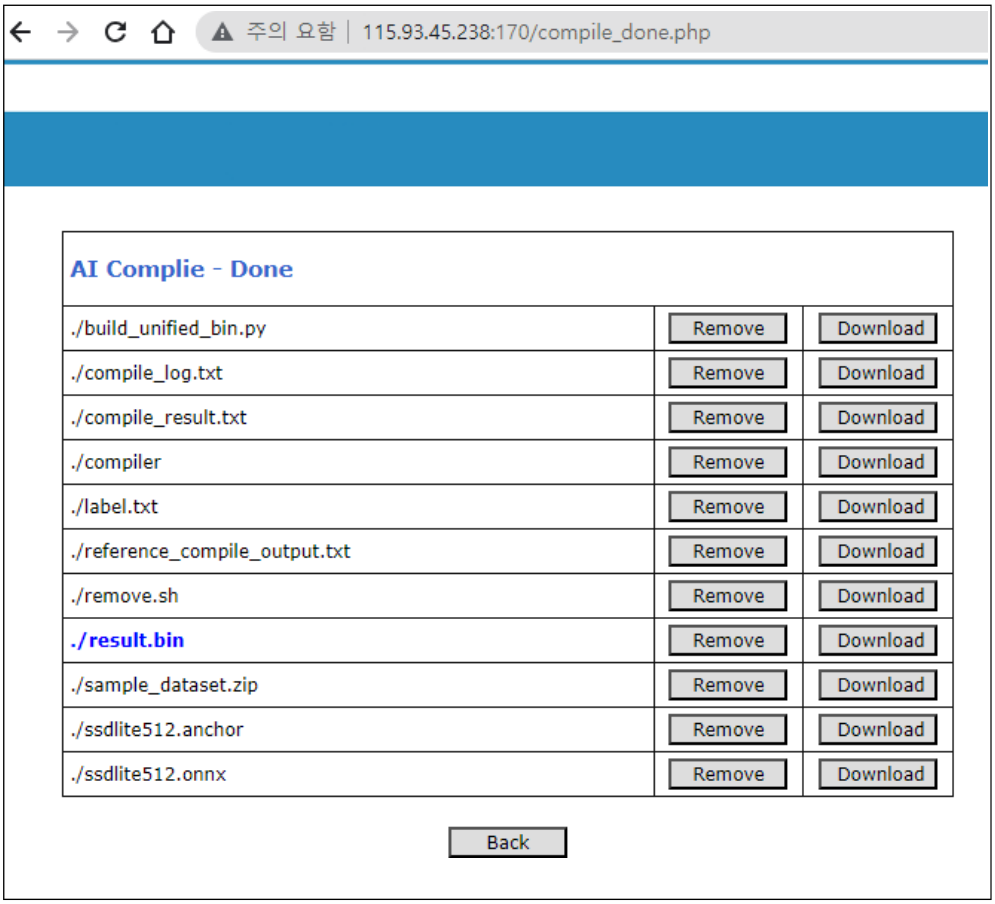


7. Progress bar shows the progress of compilation.

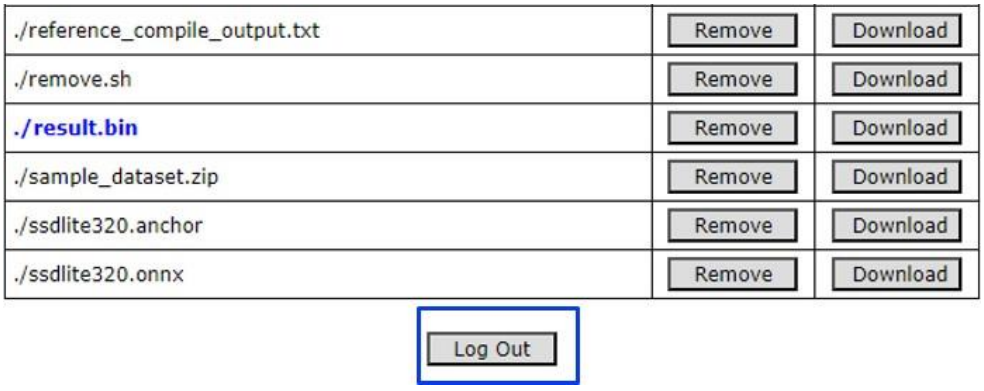


8. Compilation is complete. The ai model binaries available on En675 are highlighted in blue text. It can be downloaded and used in en675.

9. You can delete the uploaded file on http://115.93.45.238:170/compile_done.php page.



10. After downloading the result file and deleting the uploaded files, click the Log Out button to log out.



1.8. Create a compressed file (tar.gz) to upload to the En675

1. Prepare the result.bin file and label.txt file created through the above process.
2. Change the name of result.bin to use in IP camera.
   Name convention: <xxx>_<AiModelName>.bin (x: number)

   (This manual uses the name 255_ka.bin.)

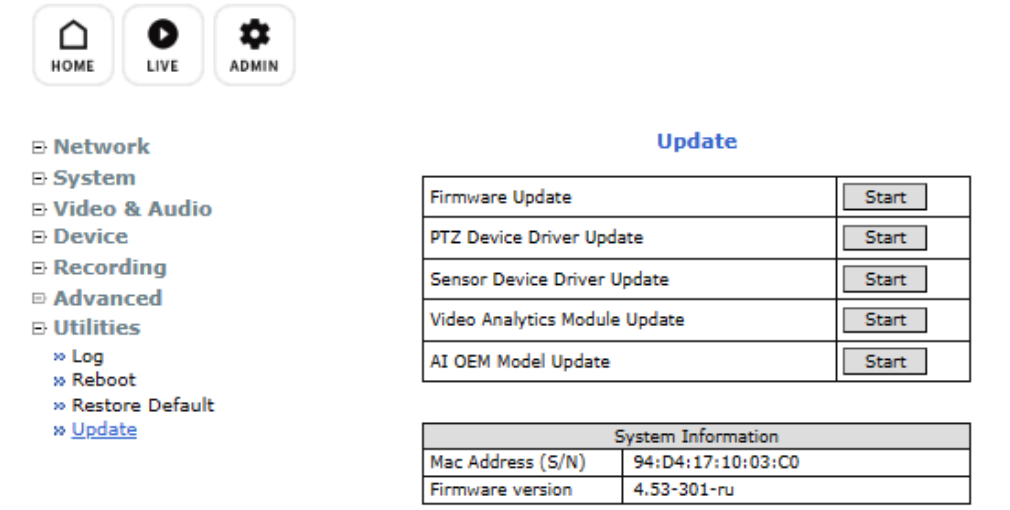
3. Give the label.txt file the same name as the binary name changed above, change the file extension to .tab, and change the contents as shown below..

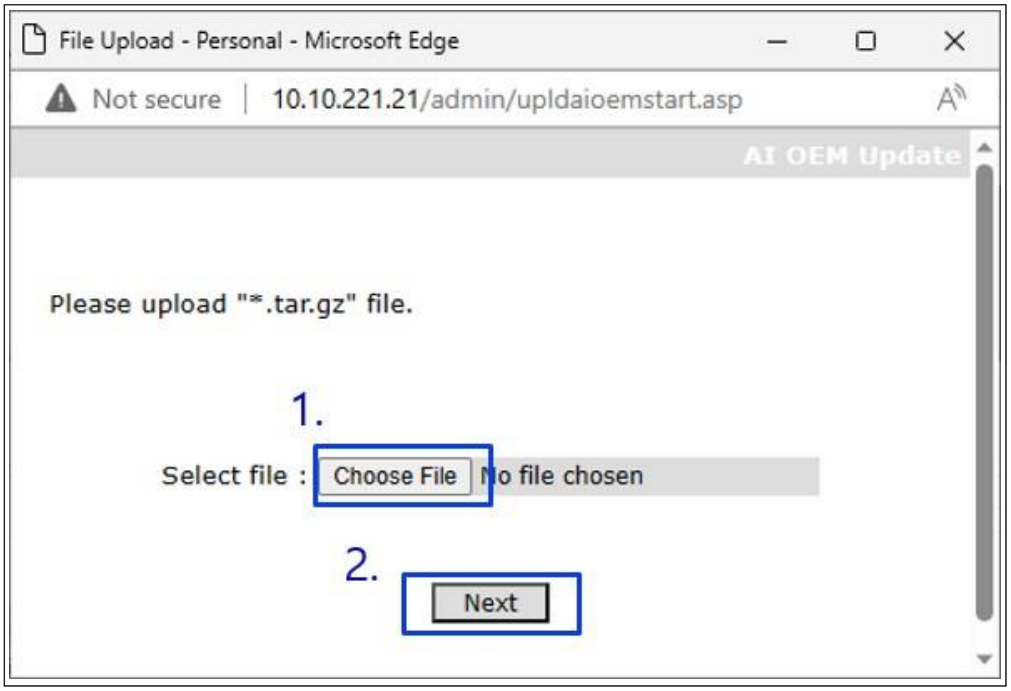
label.txt (Before)	255_ka.tab (after)
car	1=car
person	2=person
motorbike	3=motorbike
plate	4=plate

4. When the 255_ka.bin and 255_ka.tab files are ready, compress them with the command below.
   (Use the cmd window in Windows, and the default terminal in Linux.)

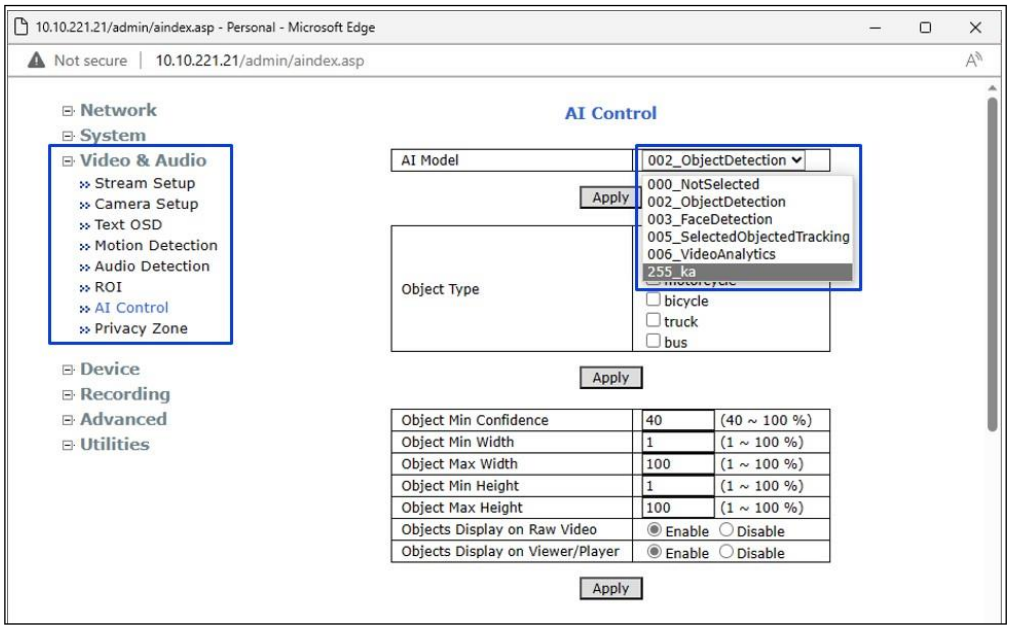
```
tar -zcvf 255_ka.tar.gz 255_ka.bin 255_ka.tab
```

5. Upload the 255_ka.tar.gz file as shown below.
   Webadmin -> Utilities -> Update -> AI OEM Update



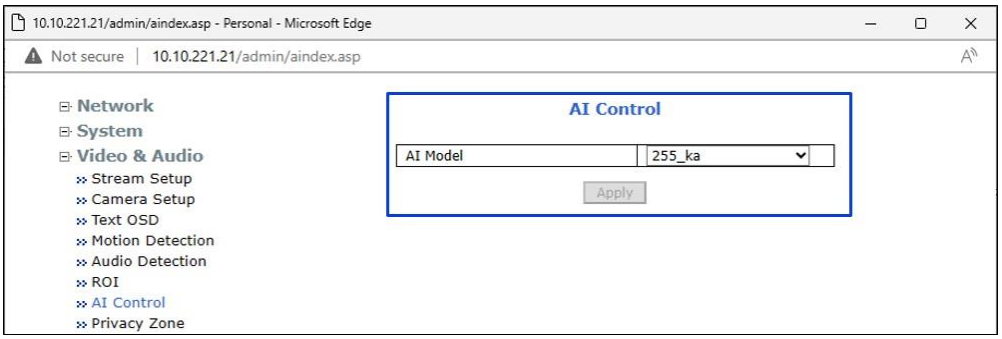


6. You can check the uploaded 255_ka ai model as shown below.
   Webadmin -> Video & Audio -> AI Control

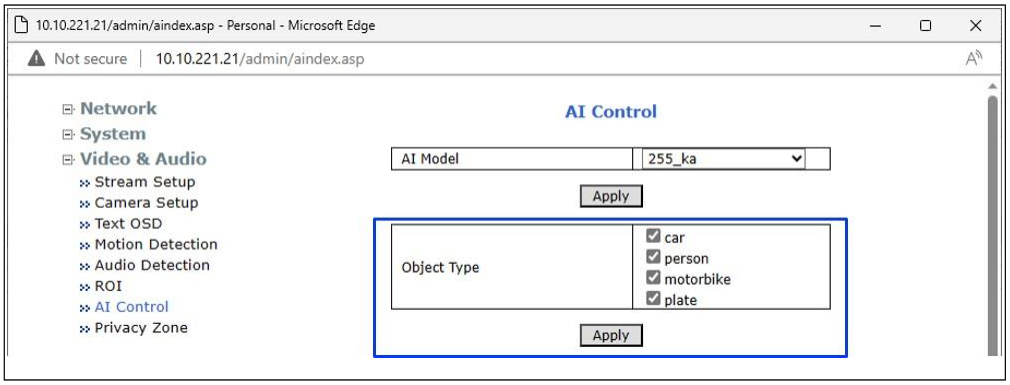


You can check the updated '255_ka' model in the AI Model drop-down box.

Select the '255_ka' model and press the 'Apply' button.



The AI model takes a few seconds to apply.



You can check the object type corresponding to the 255_ka model.

Select the checkbox of the object you want to detect and press the 'Apply' button.

7. You can check whether the ai is working through liveview.