In [ ]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6
qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%
b&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a%2f%2fwww.googleapis.
2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20https%3a%2f%2fww
ogleapis.com%2fauth%2fpeopleapi.readonly&response_type=code

Enter your authorization code:
··········
Mounted at /content/drive
```

In [ ]:

```
!nvidia-smi
```

```
Mon Aug 31 09:30:48 2020
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.66       Driver Version: 418.67       CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   40C    P8     9W /  70W |      0MiB / 15079MiB |      0%      Default |
|                               |                      |                 ERR! |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

# Using AUTOML

In [ ]:

```
!pip install --upgrade mxnet-cu100
!pip install autogluon
```

In [ ]:

```
import autogluon as ag
from autogluon import ObjectDetection as task
```

In [ ]:

```
import os
data_root = '/content/drive/My Drive/CAR DETECTION/Car_detection_autogluon'
dataset_train = task.Dataset(data_root, classes=('Car'))
```

```
>>> create dataset(VOC format)
```

# Displaying the Images

In [ ]:

```
from IPython.display import Image
Image(os.path.join(data_root,"/content/drive/My Drive/CAR
DETECTION/Car_detection_autogluon/JPEGImages/000000001.jpg"))
```

Out[ ]:

## Test Image 2:

In [ ]:

```
Image(os.path.join(data_root,"/content/drive/My Drive/CAR
DETECTION/Car_detection_autogluon/JPEGImages/000000007.jpg"))
```

Out[ ]:

## Training the model for 3 hours

In [ ]:

```
time_limits = 10800
epochs = 25
detector = task.fit(dataset_train,
                    num_trials=1,
                    epochs=epochs,
                    lr=ag.Categorical(1e-4),
                    ngpus_per_trial=1,
                    time_limits=time_limits)
```

```
scheduler_options: Key 'training_history_callback_delta_secs': Imputing default value 60
scheduler_options: Key 'delay_get_config': Imputing default value True

Starting Experiments
Num of Finished Tasks is 0
Num of Pending Tasks is 1
```

```
scheduler: FIFOScheduler(
DistributedResourceManager{
(Remote: Remote REMOTE_ID: 0,
 <Remote: 'inproc://172.28.0.2/1317/1' processes=1 threads=2, memory=13.65 GB>, Resource:
NodeResourceManager(2 CPUs, 1 GPUs))
})
```

```
Time out (secs) is 20
{'meta_arch': 'yolo3', 'dataset': <autogluon.task.object_detection.dataset.voc.CustomVOCDetection
object at 0x7fd366d945f8>, 'net': 'mobilenet1.0', 'lr': 0.0001, 'loss':
SoftmaxCrossEntropyLoss(batch_axis=0, w=None), 'num_gpus': 1, 'batch_size': 16, 'split_ratio': 0.8
, 'epochs': 25, 'num_workers': 2, 'hybridize': True, 'verbose': False, 'final_fit': False, 'seed':
223, 'data_shape': 416, 'start_epoch': 0, 'transfer': 'coco', 'lr_mode': 'step', 'lr_decay': 0.1,
'lr_decay_period': 0, 'lr_decay_epoch': '160,180', 'warmup_lr': 0.0, 'warmup_epochs': 2,
'warmup_iters': 1000, 'warmup_factor': 0.3333333333333333, 'momentum': 0.9, 'wd': 0.0005,
'log_interval': 100, 'save_prefix': 'yolo3_mobilenet1.0_custom', 'save_interval': 10,
'val_interval': 1, 'num_samples': -1, 'no_random_shape': False, 'no_wd': False, 'mixup': False, 'n
o_mixup_epochs': 20, 'label_smooth': False, 'resume': '', 'syncbn': False, 'reuse_pred_weights': T
rue, 'task_id': 7}
[Epoch 0] Training cost: 50.607,
ObjLoss=651.569,BoxCenterLoss=2.970,BoxScaleLoss=1.971,ClassLoss=0.802
[Epoch 0] Validation: nfpa=0.0 mAP=0.0
[Epoch 1] Training cost: 45.066,
ObjLoss=12.056,BoxCenterLoss=2.897,BoxScaleLoss=1.393,ClassLoss=0.660
[Epoch 1] Validation: nfpa=0.279315004513462 mAP=0.279315004513462
[Epoch 2] Training cost: 44.142,
ObjLoss=5.430,BoxCenterLoss=2.899,BoxScaleLoss=1.161,ClassLoss=0.574
[Epoch 2] Validation: nfpa=0.34489115350359373 mAP=0.34489115350359373
[Epoch 3] Training cost: 49.193,
ObjLoss=4.574,BoxCenterLoss=2.904,BoxScaleLoss=1.007,ClassLoss=0.464
[Epoch 3] Validation: nfpa=0.3813797627648872 mAP=0.3813797627648872
[Epoch 4] Training cost: 40.006,
ObjLoss=4.209,BoxCenterLoss=2.796,BoxScaleLoss=0.926,ClassLoss=0.421
[Epoch 4] Validation: nfpa=0.5014854018675692 mAP=0.5014854018675692
[Epoch 5] Training cost: 44.004,
ObjLoss=4.283,BoxCenterLoss=2.825,BoxScaleLoss=1.010,ClassLoss=0.402
[Epoch 5] Validation: nfpa=0.3321141541765857 mAP=0.3321141541765857
[Epoch 6] Training cost: 38.485,
ObjLoss=3.395,BoxCenterLoss=2.824,BoxScaleLoss=0.934,ClassLoss=0.352
[Epoch 6] Validation: nfpa=0.5160364285405915 mAP=0.5160364285405915
[Epoch 7] Training cost: 41.867,
ObjLoss=3.178,BoxCenterLoss=2.818,BoxScaleLoss=1.003,ClassLoss=0.328
[Epoch 7] Validation: nfpa=0.6095522203568247 mAP=0.6095522203568247
[Epoch 8] Training cost: 47.598,
ObjLoss=3.073,BoxCenterLoss=2.796,BoxScaleLoss=0.980,ClassLoss=0.257
[Epoch 8] Validation: nfpa=0.5690880125534651 mAP=0.5690880125534651
[Epoch 9] Training cost: 44.024,
ObjLoss=3.036,BoxCenterLoss=2.740,BoxScaleLoss=0.898,ClassLoss=0.252
[Epoch 9] Validation: nfpa=0.5382082701987864 mAP=0.5382082701987864
[Epoch 10] Training cost: 42.004,
ObjLoss=2.944,BoxCenterLoss=2.767,BoxScaleLoss=1.071,ClassLoss=0.256
[Epoch 10] Validation: nfpa=0.5859270932933749 mAP=0.5859270932933749
[Epoch 11] Training cost: 50.063,
ObjLoss=2.663,BoxCenterLoss=2.781,BoxScaleLoss=1.021,ClassLoss=0.215
[Epoch 11] Validation: nfpa=0.6533917892408458 mAP=0.6533917892408458
[Epoch 12] Training cost: 50.618,
ObjLoss=2.706,BoxCenterLoss=2.690,BoxScaleLoss=0.871,ClassLoss=0.180
[Epoch 12] Validation: nfpa=0.5736453034505683 mAP=0.5736453034505683
[Epoch 13] Training cost: 39.942,
ObjLoss=2.302,BoxCenterLoss=2.713,BoxScaleLoss=0.905,ClassLoss=0.184
[Epoch 13] Validation: nfpa=0.6110051576260681 mAP=0.6110051576260681
[Epoch 14] Training cost: 36.214,
ObjLoss=2.432,BoxCenterLoss=2.635,BoxScaleLoss=0.905,ClassLoss=0.182
[Epoch 14] Validation: nfpa=0.5791711545159821 mAP=0.5791711545159821
[Epoch 15] Training cost: 48.554,
ObjLoss=2.689,BoxCenterLoss=2.652,BoxScaleLoss=0.802,ClassLoss=0.144
[Epoch 15] Validation: nfpa=0.572981788997395 mAP=0.572981788997395
[Epoch 16] Training cost: 43.357,
ObjLoss=2.577,BoxCenterLoss=2.699,BoxScaleLoss=0.763,ClassLoss=0.164
[Epoch 16] Validation: nfpa=0.6357438292442843 mAP=0.6357438292442843
[Epoch 17] Training cost: 35.992,
ObjLoss=2.234,BoxCenterLoss=2.673,BoxScaleLoss=0.819,ClassLoss=0.136
[Epoch 17] Validation: nfpa=0.6564812990622821 mAP=0.6564812990622821
[Epoch 18] Training cost: 44.280,
ObjLoss=2.221,BoxCenterLoss=2.714,BoxScaleLoss=0.747,ClassLoss=0.130
[Epoch 18] Validation: nfpa=0.6862217249873355 mAP=0.6862217249873355
[Epoch 19] Training cost: 46.506,
ObjLoss=2.454,BoxCenterLoss=2.627,BoxScaleLoss=0.774,ClassLoss=0.106
[Epoch 19] Validation: nfpa=0.6898409682852408 mAP=0.6898409682852408
[Epoch 20] Training cost: 46.704,
ObjLoss=2.476,BoxCenterLoss=2.731,BoxScaleLoss=0.785,ClassLoss=0.133
```

```
ObjLoss=2.470,BoxCenterLoss=2.731,BoxScaleLoss=0.765,ClassLoss=0.135
[Epoch 20] Validation: nfpa=0.6543788136727497 mAP=0.6543788136727497
[Epoch 21] Training cost: 41.975,
ObjLoss=2.105,BoxCenterLoss=2.619,BoxScaleLoss=0.749,ClassLoss=0.111
[Epoch 21] Validation: nfpa=0.7475364874879737 mAP=0.7475364874879737
[Epoch 22] Training cost: 42.778,
ObjLoss=2.295,BoxCenterLoss=2.619,BoxScaleLoss=0.767,ClassLoss=0.116
[Epoch 22] Validation: nfpa=0.6920790387515018 mAP=0.6920790387515018
[Epoch 23] Training cost: 55.430,
ObjLoss=2.006,BoxCenterLoss=2.609,BoxScaleLoss=0.712,ClassLoss=0.079
[Epoch 23] Validation: nfpa=0.6945385474524668 mAP=0.6945385474524668
[Epoch 24] Training cost: 49.793,
ObjLoss=1.777,BoxCenterLoss=2.603,BoxScaleLoss=0.718,ClassLoss=0.093
[Epoch 24] Validation: nfpa=0.7192331090553358 mAP=0.7192331090553358
```

## Calculating Test Accuracy of Model

In [ ]:

```
dataset_test = task.Dataset(data_root, index_file_name='test', classes=('nfpa',))
test_map = detector.evaluate(dataset_test)
print("mAP on test dataset: {}".format(test_map[1][1]))
```

```
>>> create dataset(VOC format)
```

```
mAP on test dataset: 0.8111888111888113
```

## Saving the weights

In [ ]:

```
savefile = 'model.pkl'
detector.save(savefile)
from autogluon import Detector
new_detector = Detector.load(savefile)
```

In [ ]:

```
!ls
```

```
checkpoint     model.pkl  yolo3_mobilenet1.0_custom_train.log
dask-worker-space   sample_data
```

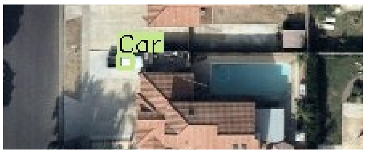
## Test Prediction 1:

In [ ]:

```
boxes = model.Predict("obj_satellite_car_pool_trained/image1.jpg",
                    vis_thresh=0.4, output_img="output.jpg")
from IPython.display import Image
Image(filename='output.jpg')
```

```
/usr/local/lib/python3.6/dist-packages/torch/nn/functional.py:2506: UserWarning: Default
upsampling behavior when mode=bilinear is changed to align_corners=False since 0.4.0. Please
specify align_corners=True if the old behavior is desired. See the documentation of nn.Upsample fo
r details.
  "See the documentation of nn.Upsample for details.".format(mode))
```

Out[ ]:



## Test Prediction 2:

In [ ]:

```
from IPython.display import Image
Image(filename='/content/drive/My Drive/CAR
DETECTION/Car_detection_autogluon/JPEGImages/000001288.jpg')
```

Out[ ]:

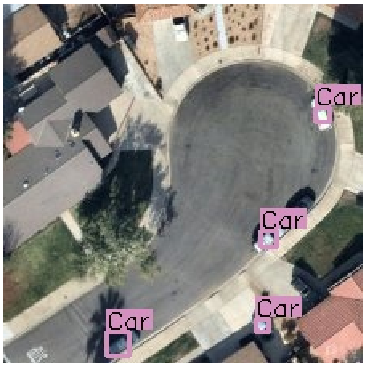
## Output:

In [ ]:

```
boxes = model.Predict("/content/drive/My Drive/CAR
DETECTION/Car_detection_autogluon/JPEGImages/000001288.jpg",
                    vis_thresh=0.4, output_img="output.jpg")
from IPython.display import Image
Image(filename='output.jpg')
```

```
/usr/local/lib/python3.6/dist-packages/torch/nn/functional.py:2506: UserWarning: Default
upsampling behavior when mode=bilinear is changed to align_corners=False since 0.4.0. Please
specify align_corners=True if the old behavior is desired. See the documentation of nn.Upsample fo
r details.
  "See the documentation of nn.Upsample for details.".format(mode))
```

Out[ ]:



## Test Image 3:

In [ ]:

```
from IPython.display import Image
Image(filename='/content/drive/My Drive/CAR
DETECTION/Car_detection_autogluon/JPEGImages/000001333.jpg')
```

Out[ ]:

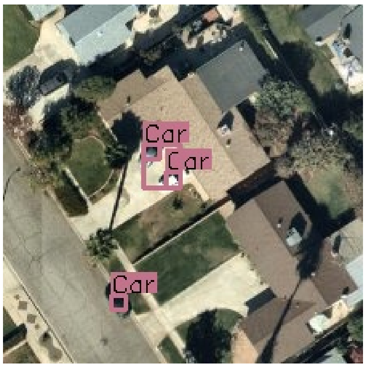
## Predicted Image

In [ ]:

```
boxes = model.Predict("/content/drive/My Drive/CAR
DETECTION/Car_detection_autogluon/JPEGImages/000001333.jpg",
                    vis_thresh=0.4, output_img="output.jpg")
from IPython.display import Image
Image(filename='output.jpg')
```

```
/usr/local/lib/python3.6/dist-packages/torch/nn/functional.py:2506: UserWarning: Default
upsampling behavior when mode=bilinear is changed to align_corners=False since 0.4.0. Please
specify align_corners=True if the old behavior is desired. See the documentation of nn.Upsample fo
r details.
  "See the documentation of nn.Upsample for details.".format(mode))
```

Out[ ]:



## Test Image 4:

In [ ]:

```
from IPython.display import Image
Image(filename='/content/drive/My Drive/CAR
DETECTION/Car_detection_autogluon/JPEGImages/000000040.jpg')
```

Out[ ]:

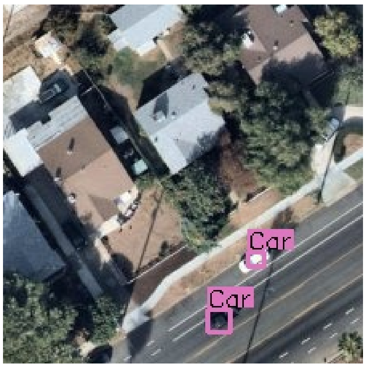
## Predicted Image

In [ ]:

```
boxes = model.Predict("/content/drive/My Drive/CAR
DETECTION/Car_detection_autogluon/JPEGImages/000000040.jpg",
                    vis_thresh=0.4, output_img="output.jpg")
from IPython.display import Image
Image(filename='output.jpg')
```

```
/usr/local/lib/python3.6/dist-packages/torch/nn/functional.py:2506: UserWarning: Default
upsampling behavior when mode=bilinear is changed to align_corners=False since 0.4.0. Please
specify align_corners=True if the old behavior is desired. See the documentation of nn.Upsample fo
r details.
  "See the documentation of nn.Upsample for details.".format(mode))
```

Out[ ]:

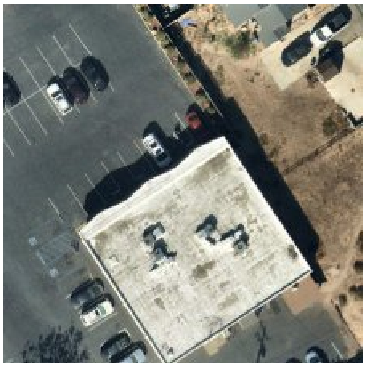


## Test Image 5:

In [ ]:

```
from IPython.display import Image
Image(filename='/content/drive/My Drive/CAR
DETECTION/Car_detection_autogluon/JPEGImages/000000009.jpg')
```

Out[ ]:



## Predicted Image 5

In [ ]:

```
boxes = model.Predict("/content/drive/My Drive/CAR
DETECTION/Car_detection_autogluon/JPEGImages/000000009.jpg",
                    vis_thresh=0.4, output_img="output.jpg")
from IPython.display import Image
Image(filename='output.jpg')
```

```
/usr/local/lib/python3.6/dist-packages/torch/nn/functional.py:2506: UserWarning: Default
upsampling behavior when mode=bilinear is changed to align_corners=False since 0.4.0. Please
specify align_corners=True if the old behavior is desired. See the documentation of nn.Upsample fo
r details.
  "See the documentation of nn.Upsample for details.".format(mode))
```

Out[ ]: