# MP3_P2

April 2, 2024

[1]:
```
# you will be prompted with a window asking to grant permissions
from google.colab import drive
drive.mount("/content/drive")
```

Mounted at /content/drive

[2]:
```
# fill in the path in your Google Drive in the string below. Note: do not␣
↪escape slashes or spaces
import os
datadir = "./"
if not os.path.exists(datadir):
  !ln -s "/content/drive/My Drive/Your/Assignment3/path/" $datadir # TODO: Fill␣
↪your Assignment 3 path
os.chdir(datadir)
!pwd
```

/content

[ ]:
```
# In a terminal, Run the `download_data.sh` script in the data folder of␣
↪assignment 3 part 2
```

[ ]:
```
import os
import random

import cv2
import numpy as np

import torch
from torch.utils.data import DataLoader
from torchvision import models

from src.resnet_yolo import resnet50
from yolo_loss import YoloLoss
from src.dataset import VocDetectorDataset
from src.eval_voc import evaluate
from src.predict import predict_image
from src.config import VOC_CLASSES, COLORS
from kaggle_submission import output_submission_csv
```

```
import matplotlib.pyplot as plt
import collections

%matplotlib inline
%load_ext autoreload
%autoreload 2
```

## 0.1 Initialization

[ ]:
```
device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
print(device)
```

```
cuda:0
```

[ ]:
```
# YOLO network hyperparameters
B = 2  # number of bounding box predictions per cell
S = 14  # width/height of network output grid (larger than 7x7 from paper since
 ↪we use a different network)
```

To implement Yolo we will rely on a pretrained classifier as the backbone for our detection network. PyTorch offers a variety of models which are pretrained on ImageNet in the `torchvision.models` package. In particular, we will use the ResNet50 architecture as a base for our detector. This is different from the base architecture in the Yolo paper and also results in a different output grid size (14x14 instead of 7x7).

Models are typically pretrained on ImageNet since the dataset is very large (> 1 million images) and widely used. The pretrained model provides a very useful weight initialization for our detector, so that the network is able to learn quickly and effectively.

[ ]:
```
load_network_path = 'checkpoints/best_detector.pth'
pretrained = True

# use to load a previously trained network
if load_network_path is not None:
    print('Loading saved network from {}'.format(load_network_path))
    net = resnet50().to(device)
    net.load_state_dict(torch.load(load_network_path))
else:
    print('Load pre-trained model')
    net = resnet50(pretrained=pretrained).to(device)
```

```
Loading saved network from checkpoints/best_detector.pth
```

[ ]:
```
learning_rate = 0.001
num_epochs = 50
batch_size = 48
```

```
# Yolo loss component coefficients (as given in Yolo v1 paper)
lambda_coord = 5
lambda_noobj = 0.5
```

### 0.2 Reading Pascal Data

Since Pascal is a small dataset (5000 in train+val) we have combined the train and val splits to train our detector. This is not typically a good practice, but we will make an exception in this case to be able to get reasonable detection results with a comparatively small object detection dataset.

The train dataset loader also using a variety of data augmentation techniques including random shift, scaling, crop, and flips. Data augmentation is slightly more complicated for detection datasets since the bounding box annotations must be kept consistent throughout the transformations.

Since the output of the detector network we train is an SxSx(B*5+C), we use an encoder to convert the original bounding box coordinates into relative grid bounding box coordinates corresponding to the expected output. We also use a decoder which allows us to convert the opposite direction into image coordinate bounding boxes.

[ ]:
```
file_root_train = r"data/VOCdevkit_2007/VOC2007/JPEGImages/"
annotation_file_train = "data/voc2007.txt"

train_dataset =␣
 ↪VocDetectorDataset(root_img_dir=file_root_train,dataset_file=annotation_file_train,train=Tr
 ↪S=S)
train_loader =␣
 ↪DataLoader(train_dataset,batch_size=batch_size,shuffle=True,num_workers=2)
print('Loaded %d train images' % len(train_dataset))
```

```
Initializing dataset
Loaded 5011 train images
```

[ ]:
```
file_root_test = "data/VOCdevkit_2007/VOC2007test/JPEGImages/"
annotation_file_test = "data/voc2007test.txt"

test_dataset =␣
 ↪VocDetectorDataset(root_img_dir=file_root_test,dataset_file=annotation_file_test,train=Fals
 ↪S=S)
test_loader =␣
 ↪DataLoader(test_dataset,batch_size=batch_size,shuffle=False,num_workers=2)
print('Loaded %d test images' % len(test_dataset))
```

```
Initializing dataset
Loaded 4950 test images
```

[ ]:
```
data = train_dataset[0]
print(data)
```

```
(tensor([[[[-116., -106., -100.,  …, -108., -107., -110.],
          [-115., -103.,  -94.,  …, -104., -105., -109.],
          [-111., -112.,  -93.,  …, -104., -106., -110.],
          …,
          [ -85.,  -95.,  -93.,  …, -102., -100.,  -97.],
          [ -78.,  -88.,  -87.,  …, -103., -103.,  -99.],
          [ -74.,  -84.,  -83.,  …, -102., -100.,  -97.]],

         [[-102.,  -96.,  -98.,  …,  -97.,  -95.,  -98.],
          [-104.,  -93.,  -90.,  …,  -93.,  -93.,  -96.],
          [-103., -104.,  -83.,  …,  -92.,  -94.,  -97.],
          …,
          [ -74.,  -80.,  -74.,  …,  -80.,  -77.,  -74.],
          [ -69.,  -77.,  -70.,  …,  -81.,  -79.,  -75.],
          [ -65.,  -73.,  -66.,  …,  -80.,  -76.,  -73.]],

         [[ -86.,  -78.,  -75.,  …,  -85.,  -84.,  -87.],
          [ -88.,  -76.,  -70.,  …,  -82.,  -82.,  -87.],
          [ -91.,  -90.,  -71.,  …,  -84.,  -84.,  -90.],
          …,
          [ -67.,  -75.,  -69.,  …,  -77.,  -75.,  -72.],
          [ -63.,  -71.,  -65.,  …,  -79.,  -78.,  -74.],
          [ -59.,  -67.,  -61.,  …,  -78.,  -75.,  -72.]]]]), tensor([[[0.0000,
0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000]],

         [[0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
          [0.0000, 0.0000, 0.0000, 0.0000],
```

```
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000]],

[[0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000]],

[[0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000]],

[[0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
```

```
 [0.0000, 0.0000, 0.0000, 0.0000]],

[[0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000]],

[[0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000]],

[[0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0817, 0.8682, 0.8184, 0.8760],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000]],

[[0.0000, 0.0000, 0.0000, 0.0000],
```

```
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000]],

        [[0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000]],

        [[0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000]],

        [[0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
```

```
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000]],

        [[0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000]],

        [[0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000]]]), tensor([[[0., 0., 0.,  …, 0.,
0., 0.],
         [0., 0., 0.,  …, 0., 0., 0.],
         [0., 0., 0.,  …, 0., 0., 0.],
         …,
         [0., 0., 0.,  …, 0., 0., 0.],
         [0., 0., 0.,  …, 0., 0., 0.],
         [0., 0., 0.,  …, 0., 0., 0.]],
```

```
         [[0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.],
          …,
          [0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.]],

         [[0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.],
          …,
          [0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.]],

         …,

         [[0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.],
          …,
          [0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.]],

         [[0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.],
          …,
          [0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.]],

         [[0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.],
          …,
          [0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.],
          [0., 0., 0.,  …, 0., 0., 0.]]]]), tensor([[False, False, False, False,
False, False, False, False, False, False,
          False, False, False, False],
         [False, False, False, False, False, False, False, False, False, False,
          False, False, False, False],
         [False, False, False, False, False, False, False, False, False, False,
          False, False, False, False],
         [False, False, False, False, False, False, False, False, False, False,
```

```
         False, False, False, False],
        [False, False, False, False, False, False, False, False, False, False,
         False, False, False, False],
        [False, False, False, False, False, False, False, False, False, False,
         False, False, False, False],
        [False, False, False, False, False, False, False, False, False, False,
         False, False, False, False],
        [False, False, False, False, False, False, False, False,  True, False,
         False, False, False, False],
        [False, False, False, False, False, False, False, False, False, False,
         False, False, False, False],
        [False, False, False, False, False, False, False, False, False, False,
         False, False, False, False],
        [False, False, False, False, False, False, False, False, False, False,
         False, False, False, False],
        [False, False, False, False, False, False, False, False, False, False,
         False, False, False, False],
        [False, False, False, False, False, False, False, False, False, False,
         False, False, False, False],
        [False, False, False, False, False, False, False, False, False, False,
         False, False, False, False]]))
```

### 0.3 Set up training tools

[ ]:
```
criterion = YoloLoss(S, B, lambda_coord, lambda_noobj)
optimizer = torch.optim.SGD(net.parameters(), lr=learning_rate, momentum=0.9,␣
 ↪weight_decay=5e-4)
```

### 0.4 Train detector

[ ]:
```
best_test_loss = np.inf
learning_rate = 1e-3
for epoch in range(num_epochs):
    net.train()

    # Update learning rate late in training
    if epoch == 30 or epoch == 40:
        learning_rate /= 10.0

    for param_group in optimizer.param_groups:
        param_group['lr'] = learning_rate

    print('\n\nStarting epoch %d / %d' % (epoch + 1, num_epochs))
    print('Learning Rate for this epoch: {}'.format(learning_rate))

    total_loss = collections.defaultdict(int)
```

```
    for i, data in enumerate(train_loader):
        data = (item.to(device) for item in data)
        images, target_boxes, target_cls, has_object_map = data
        pred = net(images)
        loss_dict = criterion(pred, target_boxes, target_cls, has_object_map)
        for key in loss_dict:
            total_loss[key] += loss_dict[key].item()

        optimizer.zero_grad()
        loss_dict['total_loss'].backward()
        optimizer.step()

        if (i+1) % 50 == 0:
            outstring = 'Epoch [%d/%d], Iter [%d/%d], Loss: ' % ((epoch+1, num_epochs, i+1, len(train_loader)))
            outstring += ', '.join( "%s=%.3f" % (key[:-5], val / (i+1)) for key, val in total_loss.items() )
            print(outstring)

    # evaluate the network on the test data
    if (epoch + 1) % 5 == 0:
        test_aps = evaluate(net, test_dataset_file=annotation_file_test, img_root=file_root_test)
        print(epoch, test_aps)
    with torch.no_grad():
        test_loss = 0.0
        net.eval()
        for i, data in enumerate(test_loader):
            data = (item.to(device) for item in data)
            images, target_boxes, target_cls, has_object_map = data

            pred = net(images)
            loss_dict = criterion(pred, target_boxes, target_cls, has_object_map)
            test_loss += loss_dict['total_loss'].item()
        test_loss /= len(test_loader)

    if best_test_loss > test_loss:
        best_test_loss = test_loss
        print('Updating best test loss: %.5f' % best_test_loss)
        torch.save(net.state_dict(),'checkpoints/best_detector.pth')

    if (epoch+1) in [5, 10, 20, 30, 40]:
        torch.save(net.state_dict(),'checkpoints/detector_epoch_%d.pth' % (epoch+1))

    torch.save(net.state_dict(),'checkpoints/detector.pth')
```

```
Starting epoch 1 / 50
Learning Rate for this epoch: 0.001
Epoch [1/50], Iter [50/209], Loss: total=23.023, reg=3.690,
containing_obj=0.332, no_obj=12.304, cls=6.696
Epoch [1/50], Iter [100/209], Loss: total=14.953, reg=3.090,
containing_obj=0.452, no_obj=6.371, cls=5.040
Epoch [1/50], Iter [150/209], Loss: total=12.040, reg=2.833,
containing_obj=0.525, no_obj=4.349, cls=4.334
Epoch [1/50], Iter [200/209], Loss: total=10.444, reg=2.674,
containing_obj=0.570, no_obj=3.322, cls=3.877
Updating best test loss: 5.28874


Starting epoch 2 / 50
Learning Rate for this epoch: 0.001
Epoch [2/50], Iter [50/209], Loss: total=5.168, reg=2.059, containing_obj=0.762,
no_obj=0.197, cls=2.150
Epoch [2/50], Iter [100/209], Loss: total=5.136, reg=2.074,
containing_obj=0.779, no_obj=0.184, cls=2.099
Epoch [2/50], Iter [150/209], Loss: total=5.020, reg=2.037,
containing_obj=0.778, no_obj=0.174, cls=2.031
Epoch [2/50], Iter [200/209], Loss: total=4.918, reg=2.004,
containing_obj=0.783, no_obj=0.165, cls=1.966
Updating best test loss: 4.64379


Starting epoch 3 / 50
Learning Rate for this epoch: 0.001
Epoch [3/50], Iter [50/209], Loss: total=4.577, reg=1.916, containing_obj=0.852,
no_obj=0.124, cls=1.685
Epoch [3/50], Iter [100/209], Loss: total=4.455, reg=1.874,
containing_obj=0.838, no_obj=0.119, cls=1.623
Epoch [3/50], Iter [150/209], Loss: total=4.459, reg=1.880,
containing_obj=0.846, no_obj=0.115, cls=1.618
Epoch [3/50], Iter [200/209], Loss: total=4.385, reg=1.850,
containing_obj=0.846, no_obj=0.112, cls=1.578
Updating best test loss: 4.35423


Starting epoch 4 / 50
Learning Rate for this epoch: 0.001
Epoch [4/50], Iter [50/209], Loss: total=4.112, reg=1.788, containing_obj=0.855,
no_obj=0.095, cls=1.374
Epoch [4/50], Iter [100/209], Loss: total=4.119, reg=1.788,
containing_obj=0.869, no_obj=0.093, cls=1.369
Epoch [4/50], Iter [150/209], Loss: total=4.085, reg=1.781,
```

```
containing_obj=0.865, no_obj=0.092, cls=1.347
Epoch [4/50], Iter [200/209], Loss: total=4.072, reg=1.781,
containing_obj=0.874, no_obj=0.090, cls=1.327
Updating best test loss: 4.03567


Starting epoch 5 / 50
Learning Rate for this epoch: 0.001
Epoch [5/50], Iter [50/209], Loss: total=3.915, reg=1.737, containing_obj=0.889,
no_obj=0.085, cls=1.203
Epoch [5/50], Iter [100/209], Loss: total=3.842, reg=1.718,
containing_obj=0.878, no_obj=0.086, cls=1.160
Epoch [5/50], Iter [150/209], Loss: total=3.859, reg=1.733,
containing_obj=0.880, no_obj=0.086, cls=1.160
Epoch [5/50], Iter [200/209], Loss: total=3.809, reg=1.724,
containing_obj=0.868, no_obj=0.088, cls=1.128
---Evaluate model on test samples---
```

100%|     | 4950/4950 [01:59<00:00, 41.33it/s]

```
---class aeroplane ap 0.07658872725054766---
---class bicycle ap 0.05062845689428884---
---class bird ap 0.06031496703720878---
---class boat ap 0.015634869227697974---
---class bottle ap 0.0161706925891787---
---class bus ap 0.0--- (no predictions for this class)
---class car ap 0.12292450106729097---
---class cat ap 0.0111731843575419---
---class chair ap 0.03533659231025792---
---class cow ap 0.0---
---class diningtable ap 0.0--- (no predictions for this class)
---class dog ap 0.004703476482617587---
---class horse ap 0.05198014629049112---
---class motorbike ap 0.0--- (no predictions for this class)
---class person ap 0.04971351804509193---
---class pottedplant ap 0.009684292097146533---
---class sheep ap 0.019559228650137744---
---class sofa ap 0.0--- (no predictions for this class)
---class train ap 0.0--- (no predictions for this class)
---class tvmonitor ap 0.16219807005625944---
---map 0.03433053611778785---
4 [0.07658872725054766, 0.05062845689428884, 0.06031496703720878,
0.015634869227697974, 0.0161706925891787, 0.0, 0.12292450106729097,
0.0111731843575419, 0.03533659231025792, 0.0, 0.0, 0.004703476482617587,
0.05198014629049112, 0.0, 0.04971351804509193, 0.009684292097146533,
0.019559228650137744, 0.0, 0.0, 0.16219807005625944]
Updating best test loss: 3.77643
```

```
Starting epoch 6 / 50
Learning Rate for this epoch: 0.001
Epoch [6/50], Iter [50/209], Loss: total=3.530, reg=1.647, containing_obj=0.830,
no_obj=0.098, cls=0.954
Epoch [6/50], Iter [100/209], Loss: total=3.536, reg=1.650,
containing_obj=0.829, no_obj=0.102, cls=0.955
Epoch [6/50], Iter [150/209], Loss: total=3.550, reg=1.661,
containing_obj=0.826, no_obj=0.107, cls=0.956
Epoch [6/50], Iter [200/209], Loss: total=3.536, reg=1.648,
containing_obj=0.824, no_obj=0.110, cls=0.953
Updating best test loss: 3.60945


Starting epoch 7 / 50
Learning Rate for this epoch: 0.001
Epoch [7/50], Iter [50/209], Loss: total=3.457, reg=1.609, containing_obj=0.837,
no_obj=0.124, cls=0.886
Epoch [7/50], Iter [100/209], Loss: total=3.449, reg=1.628,
containing_obj=0.829, no_obj=0.122, cls=0.871
Epoch [7/50], Iter [150/209], Loss: total=3.389, reg=1.597,
containing_obj=0.817, no_obj=0.122, cls=0.853
Epoch [7/50], Iter [200/209], Loss: total=3.403, reg=1.600,
containing_obj=0.817, no_obj=0.124, cls=0.862
Updating best test loss: 3.55916


Starting epoch 8 / 50
Learning Rate for this epoch: 0.001
Epoch [8/50], Iter [50/209], Loss: total=3.353, reg=1.586, containing_obj=0.839,
no_obj=0.124, cls=0.803
Epoch [8/50], Iter [100/209], Loss: total=3.279, reg=1.549,
containing_obj=0.822, no_obj=0.127, cls=0.781
Epoch [8/50], Iter [150/209], Loss: total=3.265, reg=1.553,
containing_obj=0.816, no_obj=0.128, cls=0.768
Epoch [8/50], Iter [200/209], Loss: total=3.256, reg=1.551,
containing_obj=0.809, no_obj=0.128, cls=0.768
Updating best test loss: 3.41244


Starting epoch 9 / 50
Learning Rate for this epoch: 0.001
Epoch [9/50], Iter [50/209], Loss: total=3.145, reg=1.535, containing_obj=0.804,
no_obj=0.130, cls=0.676
Epoch [9/50], Iter [100/209], Loss: total=3.182, reg=1.542,
containing_obj=0.810, no_obj=0.133, cls=0.697
Epoch [9/50], Iter [150/209], Loss: total=3.187, reg=1.536,
containing_obj=0.815, no_obj=0.131, cls=0.704
Epoch [9/50], Iter [200/209], Loss: total=3.181, reg=1.537,
```

```
containing_obj=0.811, no_obj=0.131, cls=0.702
Updating best test loss: 3.35227


Starting epoch 10 / 50
Learning Rate for this epoch: 0.001
Epoch [10/50], Iter [50/209], Loss: total=3.063, reg=1.488,
containing_obj=0.799, no_obj=0.131, cls=0.645
Epoch [10/50], Iter [100/209], Loss: total=3.017, reg=1.466,
containing_obj=0.787, no_obj=0.133, cls=0.631
Epoch [10/50], Iter [150/209], Loss: total=3.039, reg=1.481,
containing_obj=0.793, no_obj=0.133, cls=0.632
Epoch [10/50], Iter [200/209], Loss: total=3.037, reg=1.478,
containing_obj=0.794, no_obj=0.134, cls=0.632
---Evaluate model on test samples---

100%|     | 4950/4950 [01:55<00:00, 42.94it/s]

---class aeroplane ap 0.29672594414934905---
---class bicycle ap 0.3422849102545965---
---class bird ap 0.26962383409735075---
---class boat ap 0.11380292495688714---
---class bottle ap 0.0465665892740073---
---class bus ap 0.25234830916967904---
---class car ap 0.4968924330943932---
---class cat ap 0.5211356780331985---
---class chair ap 0.16107418955618402---
---class cow ap 0.14420768319136462---
---class diningtable ap 0.0048543689320388345---
---class dog ap 0.44054838096319154---
---class horse ap 0.44923080267119464---
---class motorbike ap 0.3760258183215436---
---class person ap 0.2922193643238318---
---class pottedplant ap 0.06539052004378372---
---class sheep ap 0.18705394618455345---
---class sofa ap 0.16438100439114636---
---class train ap 0.45297452404630134---
---class tvmonitor ap 0.3515317055710824---
---map 0.2714436465612839---
9 [0.29672594414934905, 0.3422849102545965, 0.26962383409735075,
0.11380292495688714, 0.0465665892740073, 0.25234830916967904,
0.4968924330943932, 0.5211356780331985, 0.16107418955618402,
0.14420768319136462, 0.0048543689320388345, 0.44054838096319154,
0.44923080267119464, 0.3760258183215436, 0.2922193643238318,
0.06539052004378372, 0.18705394618455345, 0.16438100439114636,
0.45297452404630134, 0.3515317055710824]
Updating best test loss: 3.32889
```

```
Starting epoch 11 / 50
Learning Rate for this epoch: 0.001
Epoch [11/50], Iter [50/209], Loss: total=2.953, reg=1.446,
containing_obj=0.790, no_obj=0.135, cls=0.582
Epoch [11/50], Iter [100/209], Loss: total=2.943, reg=1.435,
containing_obj=0.790, no_obj=0.139, cls=0.579
Epoch [11/50], Iter [150/209], Loss: total=2.946, reg=1.444,
containing_obj=0.788, no_obj=0.140, cls=0.574
Epoch [11/50], Iter [200/209], Loss: total=2.926, reg=1.438,
containing_obj=0.780, no_obj=0.141, cls=0.567
Updating best test loss: 3.20719


Starting epoch 12 / 50
Learning Rate for this epoch: 0.001
Epoch [12/50], Iter [50/209], Loss: total=2.938, reg=1.448,
containing_obj=0.788, no_obj=0.154, cls=0.547
Epoch [12/50], Iter [100/209], Loss: total=2.832, reg=1.394,
containing_obj=0.761, no_obj=0.156, cls=0.520
Epoch [12/50], Iter [150/209], Loss: total=2.823, reg=1.391,
containing_obj=0.758, no_obj=0.156, cls=0.518
Epoch [12/50], Iter [200/209], Loss: total=2.831, reg=1.394,
containing_obj=0.759, no_obj=0.155, cls=0.524
Updating best test loss: 3.12265


Starting epoch 13 / 50
Learning Rate for this epoch: 0.001
Epoch [13/50], Iter [50/209], Loss: total=2.705, reg=1.329,
containing_obj=0.739, no_obj=0.161, cls=0.475
Epoch [13/50], Iter [100/209], Loss: total=2.816, reg=1.391,
containing_obj=0.753, no_obj=0.165, cls=0.507
Epoch [13/50], Iter [150/209], Loss: total=2.810, reg=1.392,
containing_obj=0.748, no_obj=0.167, cls=0.503
Epoch [13/50], Iter [200/209], Loss: total=2.778, reg=1.379,
containing_obj=0.737, no_obj=0.166, cls=0.496


Starting epoch 14 / 50
Learning Rate for this epoch: 0.001
Epoch [14/50], Iter [50/209], Loss: total=2.766, reg=1.355,
containing_obj=0.744, no_obj=0.166, cls=0.501
Epoch [14/50], Iter [100/209], Loss: total=2.730, reg=1.347,
containing_obj=0.730, no_obj=0.172, cls=0.482
Epoch [14/50], Iter [150/209], Loss: total=2.732, reg=1.362,
containing_obj=0.730, no_obj=0.172, cls=0.468
Epoch [14/50], Iter [200/209], Loss: total=2.706, reg=1.340,
containing_obj=0.726, no_obj=0.171, cls=0.468
```

```
Updating best test loss: 3.06067


Starting epoch 15 / 50
Learning Rate for this epoch: 0.001
Epoch [15/50], Iter [50/209], Loss: total=2.670, reg=1.348,
containing_obj=0.734, no_obj=0.171, cls=0.417
Epoch [15/50], Iter [100/209], Loss: total=2.691, reg=1.348,
containing_obj=0.731, no_obj=0.172, cls=0.440
Epoch [15/50], Iter [150/209], Loss: total=2.654, reg=1.317,
containing_obj=0.726, no_obj=0.172, cls=0.440
Epoch [15/50], Iter [200/209], Loss: total=2.636, reg=1.305,
containing_obj=0.718, no_obj=0.172, cls=0.441
---Evaluate model on test samples---
100%|     | 4950/4950 [01:49<00:00, 45.00it/s]
---class aeroplane ap 0.353683881278508---
---class bicycle ap 0.5256176577417708---
---class bird ap 0.3356674510308097---
---class boat ap 0.16124285849869038---
---class bottle ap 0.08810782975750252---
---class bus ap 0.4250292824366394---
---class car ap 0.5366674313982549---
---class cat ap 0.598304841212767---
---class chair ap 0.2232592629911557---
---class cow ap 0.2789533399214591---
---class diningtable ap 0.05270457697642164---
---class dog ap 0.5249338844330504---
---class horse ap 0.5374441023029601---
---class motorbike ap 0.42264480520659303---
---class person ap 0.40555980096429206---
---class pottedplant ap 0.11200195599409636---
---class sheep ap 0.29994128351451826---
---class sofa ap 0.31837120455312473---
---class train ap 0.60919846631995---
---class tvmonitor ap 0.35808844254523087---
---map 0.3583711179538897---
14 [0.353683881278508, 0.5256176577417708, 0.3356674510308097,
0.16124285849869038, 0.08810782975750252, 0.4250292824366394,
0.5366674313982549, 0.598304841212767, 0.2232592629911557, 0.2789533399214591,
0.05270457697642164, 0.5249338844330504, 0.5374441023029601,
0.42264480520659303, 0.40555980096429206, 0.11200195599409636,
0.29994128351451826, 0.31837120455312473, 0.60919846631995, 0.35808844254523087]
Updating best test loss: 3.04224


Starting epoch 16 / 50
Learning Rate for this epoch: 0.001
```

```
Epoch [16/50], Iter [50/209], Loss: total=2.486, reg=1.218,
containing_obj=0.691, no_obj=0.172, cls=0.405
Epoch [16/50], Iter [100/209], Loss: total=2.511, reg=1.239,
containing_obj=0.683, no_obj=0.177, cls=0.413
Epoch [16/50], Iter [150/209], Loss: total=2.532, reg=1.250,
containing_obj=0.694, no_obj=0.175, cls=0.413
Epoch [16/50], Iter [200/209], Loss: total=2.553, reg=1.266,
containing_obj=0.697, no_obj=0.174, cls=0.417
Updating best test loss: 3.01288


Starting epoch 17 / 50
Learning Rate for this epoch: 0.001
Epoch [17/50], Iter [50/209], Loss: total=2.566, reg=1.276,
containing_obj=0.713, no_obj=0.180, cls=0.397
Epoch [17/50], Iter [100/209], Loss: total=2.487, reg=1.223,
containing_obj=0.695, no_obj=0.182, cls=0.387
Epoch [17/50], Iter [150/209], Loss: total=2.475, reg=1.217,
containing_obj=0.691, no_obj=0.182, cls=0.384
Epoch [17/50], Iter [200/209], Loss: total=2.489, reg=1.221,
containing_obj=0.697, no_obj=0.180, cls=0.391
Updating best test loss: 2.97498


Starting epoch 18 / 50
Learning Rate for this epoch: 0.001
Epoch [18/50], Iter [50/209], Loss: total=2.517, reg=1.262,
containing_obj=0.710, no_obj=0.187, cls=0.358
Epoch [18/50], Iter [100/209], Loss: total=2.508, reg=1.253,
containing_obj=0.705, no_obj=0.183, cls=0.367
Epoch [18/50], Iter [150/209], Loss: total=2.484, reg=1.234,
containing_obj=0.695, no_obj=0.181, cls=0.374
Epoch [18/50], Iter [200/209], Loss: total=2.507, reg=1.244,
containing_obj=0.700, no_obj=0.181, cls=0.382
Updating best test loss: 2.89958


Starting epoch 19 / 50
Learning Rate for this epoch: 0.001
Epoch [19/50], Iter [50/209], Loss: total=2.396, reg=1.171,
containing_obj=0.688, no_obj=0.175, cls=0.362
Epoch [19/50], Iter [100/209], Loss: total=2.444, reg=1.215,
containing_obj=0.692, no_obj=0.177, cls=0.360
Epoch [19/50], Iter [150/209], Loss: total=2.421, reg=1.198,
containing_obj=0.682, no_obj=0.180, cls=0.361
Epoch [19/50], Iter [200/209], Loss: total=2.426, reg=1.201,
containing_obj=0.681, no_obj=0.182, cls=0.362
```

```
Starting epoch 20 / 50
Learning Rate for this epoch: 0.001
Epoch [20/50], Iter [50/209], Loss: total=2.345, reg=1.163,
containing_obj=0.684, no_obj=0.180, cls=0.318
Epoch [20/50], Iter [100/209], Loss: total=2.331, reg=1.150,
containing_obj=0.673, no_obj=0.182, cls=0.326
Epoch [20/50], Iter [150/209], Loss: total=2.346, reg=1.162,
containing_obj=0.679, no_obj=0.183, cls=0.322
Epoch [20/50], Iter [200/209], Loss: total=2.356, reg=1.165,
containing_obj=0.678, no_obj=0.184, cls=0.329
---Evaluate model on test samples---

100%|     | 4950/4950 [01:50<00:00, 44.98it/s]

---class aeroplane ap 0.44728433469512224---
---class bicycle ap 0.5209844377553562---
---class bird ap 0.4001112153420066---
---class boat ap 0.1782360244094975---
---class bottle ap 0.12431433578342119---
---class bus ap 0.49089998788611183---
---class car ap 0.5543484487470753---
---class cat ap 0.5593981607331366---
---class chair ap 0.22098080775915271---
---class cow ap 0.42783670584498845---
---class diningtable ap 0.17728054829978707---
---class dog ap 0.5597769708665804---
---class horse ap 0.6254283583663549---
---class motorbike ap 0.5664770219279924---
---class person ap 0.4564772852679984---
---class pottedplant ap 0.1360019135184708---
---class sheep ap 0.3841037883120979---
---class sofa ap 0.4167732522132447---
---class train ap 0.5855054093211073---
---class tvmonitor ap 0.3582455920919174---
---map 0.40952322995707097---
19 [0.44728433469512224, 0.5209844377553562, 0.4001112153420066,
0.1782360244094975, 0.12431433578342119, 0.49089998788611183,
0.5543484487470753, 0.5593981607331366, 0.22098080775915271,
0.42783670584498845, 0.17728054829978707, 0.5597769708665804,
0.6254283583663549, 0.5664770219279924, 0.4564772852679984, 0.1360019135184708,
0.3841037883120979, 0.4167732522132447, 0.5855054093211073, 0.3582455920919174]


Starting epoch 21 / 50
Learning Rate for this epoch: 0.001
Epoch [21/50], Iter [50/209], Loss: total=2.316, reg=1.158,
containing_obj=0.658, no_obj=0.179, cls=0.321
Epoch [21/50], Iter [100/209], Loss: total=2.303, reg=1.138,
```

```
containing_obj=0.668, no_obj=0.184, cls=0.312
Epoch [21/50], Iter [150/209], Loss: total=2.344, reg=1.164,
containing_obj=0.676, no_obj=0.187, cls=0.317
Epoch [21/50], Iter [200/209], Loss: total=2.339, reg=1.157,
containing_obj=0.673, no_obj=0.188, cls=0.321


Starting epoch 22 / 50
Learning Rate for this epoch: 0.001
Epoch [22/50], Iter [50/209], Loss: total=2.302, reg=1.154,
containing_obj=0.672, no_obj=0.181, cls=0.295
Epoch [22/50], Iter [100/209], Loss: total=2.287, reg=1.136,
containing_obj=0.671, no_obj=0.184, cls=0.295
Epoch [22/50], Iter [150/209], Loss: total=2.297, reg=1.145,
containing_obj=0.675, no_obj=0.184, cls=0.293
Epoch [22/50], Iter [200/209], Loss: total=2.313, reg=1.146,
containing_obj=0.677, no_obj=0.186, cls=0.304
Updating best test loss: 2.86011


Starting epoch 23 / 50
Learning Rate for this epoch: 0.001
Epoch [23/50], Iter [50/209], Loss: total=2.217, reg=1.090,
containing_obj=0.643, no_obj=0.185, cls=0.299
Epoch [23/50], Iter [100/209], Loss: total=2.229, reg=1.098,
containing_obj=0.640, no_obj=0.187, cls=0.304
Epoch [23/50], Iter [150/209], Loss: total=2.212, reg=1.084,
containing_obj=0.642, no_obj=0.188, cls=0.297
Epoch [23/50], Iter [200/209], Loss: total=2.229, reg=1.088,
containing_obj=0.648, no_obj=0.190, cls=0.302


Starting epoch 24 / 50
Learning Rate for this epoch: 0.001
Epoch [24/50], Iter [50/209], Loss: total=2.119, reg=1.043,
containing_obj=0.638, no_obj=0.194, cls=0.244
Epoch [24/50], Iter [100/209], Loss: total=2.162, reg=1.061,
containing_obj=0.639, no_obj=0.193, cls=0.268
Epoch [24/50], Iter [150/209], Loss: total=2.198, reg=1.079,
containing_obj=0.647, no_obj=0.193, cls=0.280
Epoch [24/50], Iter [200/209], Loss: total=2.201, reg=1.083,
containing_obj=0.647, no_obj=0.193, cls=0.278
Updating best test loss: 2.82955


Starting epoch 25 / 50
Learning Rate for this epoch: 0.001
Epoch [25/50], Iter [50/209], Loss: total=2.209, reg=1.079,
```

```
containing_obj=0.650, no_obj=0.195, cls=0.286
Epoch [25/50], Iter [100/209], Loss: total=2.174, reg=1.066,
containing_obj=0.640, no_obj=0.194, cls=0.274
Epoch [25/50], Iter [150/209], Loss: total=2.181, reg=1.071,
containing_obj=0.646, no_obj=0.193, cls=0.271
Epoch [25/50], Iter [200/209], Loss: total=2.164, reg=1.061,
containing_obj=0.641, no_obj=0.194, cls=0.268
---Evaluate model on test samples---

100%|     | 4950/4950 [02:02<00:00, 40.43it/s]

---class aeroplane ap 0.5149458020731885---
---class bicycle ap 0.5106475588148521---
---class bird ap 0.46117354577234015---
---class boat ap 0.219702083376801---
---class bottle ap 0.15645203304209096---
---class bus ap 0.5717432535826358---
---class car ap 0.6270526877149574---
---class cat ap 0.6405853965777057---
---class chair ap 0.2519842755268026---
---class cow ap 0.35256324374888814---
---class diningtable ap 0.21343934006471996---
---class dog ap 0.5578261301333471---
---class horse ap 0.6606943988185616---
---class motorbike ap 0.5226537682252406---
---class person ap 0.48934373545790183---
---class pottedplant ap 0.15633199963217598---
---class sheep ap 0.4387286403033152---
---class sofa ap 0.45115321023353133---
---class train ap 0.6454942618790018---
---class tvmonitor ap 0.43682795137218233---
---map 0.44396716581751205---
24 [0.5149458020731885, 0.5106475588148521, 0.46117354577234015,
0.219702083376801, 0.15645203304209096, 0.5717432535826358, 0.6270526877149574,
0.6405853965777057, 0.2519842755268026, 0.35256324374888814,
0.21343934006471996, 0.5578261301333471, 0.6606943988185616, 0.5226537682252406,
0.48934373545790183, 0.15633199963217598, 0.4387286403033152,
0.45115321023353133, 0.6454942618790018, 0.43682795137218233]


Starting epoch 26 / 50
Learning Rate for this epoch: 0.001
Epoch [26/50], Iter [50/209], Loss: total=2.261, reg=1.096,
containing_obj=0.684, no_obj=0.192, cls=0.289
Epoch [26/50], Iter [100/209], Loss: total=2.158, reg=1.043,
containing_obj=0.649, no_obj=0.196, cls=0.269
Epoch [26/50], Iter [150/209], Loss: total=2.130, reg=1.033,
containing_obj=0.637, no_obj=0.197, cls=0.263
Epoch [26/50], Iter [200/209], Loss: total=2.133, reg=1.036,
```

```
containing_obj=0.638, no_obj=0.194, cls=0.264


Starting epoch 27 / 50
Learning Rate for this epoch: 0.001
Epoch [27/50], Iter [50/209], Loss: total=2.077, reg=1.036,
containing_obj=0.606, no_obj=0.193, cls=0.243
Epoch [27/50], Iter [100/209], Loss: total=2.110, reg=1.041,
containing_obj=0.617, no_obj=0.192, cls=0.259
Epoch [27/50], Iter [150/209], Loss: total=2.112, reg=1.029,
containing_obj=0.623, no_obj=0.196, cls=0.265
Epoch [27/50], Iter [200/209], Loss: total=2.096, reg=1.021,
containing_obj=0.617, no_obj=0.197, cls=0.261
Updating best test loss: 2.81818


Starting epoch 28 / 50
Learning Rate for this epoch: 0.001
Epoch [28/50], Iter [50/209], Loss: total=2.155, reg=1.037,
containing_obj=0.642, no_obj=0.188, cls=0.288
Epoch [28/50], Iter [100/209], Loss: total=2.093, reg=1.023,
containing_obj=0.620, no_obj=0.192, cls=0.258
Epoch [28/50], Iter [150/209], Loss: total=2.103, reg=1.025,
containing_obj=0.622, no_obj=0.194, cls=0.262
Epoch [28/50], Iter [200/209], Loss: total=2.100, reg=1.029,
containing_obj=0.621, no_obj=0.195, cls=0.255
Updating best test loss: 2.80208


Starting epoch 29 / 50
Learning Rate for this epoch: 0.001
Epoch [29/50], Iter [50/209], Loss: total=2.043, reg=0.984,
containing_obj=0.614, no_obj=0.198, cls=0.247
Epoch [29/50], Iter [100/209], Loss: total=2.067, reg=1.005,
containing_obj=0.620, no_obj=0.199, cls=0.243
Epoch [29/50], Iter [150/209], Loss: total=2.059, reg=1.001,
containing_obj=0.620, no_obj=0.198, cls=0.240
Epoch [29/50], Iter [200/209], Loss: total=2.047, reg=0.996,
containing_obj=0.614, no_obj=0.198, cls=0.240
Updating best test loss: 2.79635


Starting epoch 30 / 50
Learning Rate for this epoch: 0.001
Epoch [30/50], Iter [50/209], Loss: total=2.055, reg=1.001,
containing_obj=0.617, no_obj=0.200, cls=0.237
Epoch [30/50], Iter [100/209], Loss: total=2.030, reg=0.992,
containing_obj=0.613, no_obj=0.196, cls=0.228
```

```
Epoch [30/50], Iter [150/209], Loss: total=2.047, reg=1.001,
containing_obj=0.617, no_obj=0.195, cls=0.234
Epoch [30/50], Iter [200/209], Loss: total=2.046, reg=0.998,
containing_obj=0.617, no_obj=0.196, cls=0.235
---Evaluate model on test samples---

100%|     | 4950/4950 [02:16<00:00, 36.25it/s]

---class aeroplane ap 0.4719819809800481---
---class bicycle ap 0.5780565318334638---
---class bird ap 0.4494709722559183---
---class boat ap 0.2924411458666887---
---class bottle ap 0.17512803769885577---
---class bus ap 0.562238838562554---
---class car ap 0.6349132255562765---
---class cat ap 0.6546596536208156---
---class chair ap 0.2358103752066274---
---class cow ap 0.34905603398848234---
---class diningtable ap 0.2802260169057064---
---class dog ap 0.5733626063495525---
---class horse ap 0.6389986491725825---
---class motorbike ap 0.5315041878862564---
---class person ap 0.49390194785564623---
---class pottedplant ap 0.15866587891490264---
---class sheep ap 0.4187775294345249---
---class sofa ap 0.42169926734926033---
---class train ap 0.63268485842851---
---class tvmonitor ap 0.47652801270692446---
---map 0.45150528752867986---
29 [0.4719819809800481, 0.5780565318334638, 0.4494709722559183,
0.2924411458666887, 0.17512803769885577, 0.562238838562554, 0.6349132255562765,
0.6546596536208156, 0.2358103752066274, 0.34905603398848234, 0.2802260169057064,
0.5733626063495525, 0.6389986491725825, 0.5315041878862564, 0.49390194785564623,
0.15866587891490264, 0.4187775294345249, 0.42169926734926033, 0.63268485842851,
0.47652801270692446]
Updating best test loss: 2.76395


Starting epoch 31 / 50
Learning Rate for this epoch: 0.0001
Epoch [31/50], Iter [50/209], Loss: total=2.007, reg=0.987,
containing_obj=0.607, no_obj=0.189, cls=0.225
Epoch [31/50], Iter [100/209], Loss: total=1.956, reg=0.964,
containing_obj=0.594, no_obj=0.189, cls=0.209
Epoch [31/50], Iter [150/209], Loss: total=1.953, reg=0.956,
containing_obj=0.601, no_obj=0.189, cls=0.206
Epoch [31/50], Iter [200/209], Loss: total=1.942, reg=0.950,
containing_obj=0.599, no_obj=0.192, cls=0.202
Updating best test loss: 2.70105
```

```
Starting epoch 32 / 50
Learning Rate for this epoch: 0.0001
Epoch [32/50], Iter [50/209], Loss: total=1.856, reg=0.879,
containing_obj=0.588, no_obj=0.203, cls=0.187
Epoch [32/50], Iter [100/209], Loss: total=1.893, reg=0.902,
containing_obj=0.595, no_obj=0.203, cls=0.194
Epoch [32/50], Iter [150/209], Loss: total=1.906, reg=0.912,
containing_obj=0.596, no_obj=0.202, cls=0.198
Epoch [32/50], Iter [200/209], Loss: total=1.892, reg=0.903,
containing_obj=0.593, no_obj=0.200, cls=0.195
Updating best test loss: 2.69919


Starting epoch 33 / 50
Learning Rate for this epoch: 0.0001
Epoch [33/50], Iter [50/209], Loss: total=1.886, reg=0.912,
containing_obj=0.577, no_obj=0.200, cls=0.196
Epoch [33/50], Iter [100/209], Loss: total=1.835, reg=0.886,
containing_obj=0.566, no_obj=0.202, cls=0.181
Epoch [33/50], Iter [150/209], Loss: total=1.847, reg=0.890,
containing_obj=0.572, no_obj=0.203, cls=0.182
Epoch [33/50], Iter [200/209], Loss: total=1.860, reg=0.894,
containing_obj=0.578, no_obj=0.201, cls=0.187
Updating best test loss: 2.68816


Starting epoch 34 / 50
Learning Rate for this epoch: 0.0001
Epoch [34/50], Iter [50/209], Loss: total=1.796, reg=0.854,
containing_obj=0.563, no_obj=0.202, cls=0.177
Epoch [34/50], Iter [100/209], Loss: total=1.819, reg=0.869,
containing_obj=0.564, no_obj=0.200, cls=0.186
Epoch [34/50], Iter [150/209], Loss: total=1.840, reg=0.881,
containing_obj=0.576, no_obj=0.202, cls=0.181
Epoch [34/50], Iter [200/209], Loss: total=1.856, reg=0.890,
containing_obj=0.582, no_obj=0.200, cls=0.183


Starting epoch 35 / 50
Learning Rate for this epoch: 0.0001
Epoch [35/50], Iter [50/209], Loss: total=1.840, reg=0.864,
containing_obj=0.585, no_obj=0.207, cls=0.183
Epoch [35/50], Iter [100/209], Loss: total=1.877, reg=0.904,
containing_obj=0.586, no_obj=0.205, cls=0.183
Epoch [35/50], Iter [150/209], Loss: total=1.850, reg=0.881,
containing_obj=0.583, no_obj=0.205, cls=0.181
```

```
Epoch [35/50], Iter [200/209], Loss: total=1.835, reg=0.874,
containing_obj=0.577, no_obj=0.205, cls=0.179
---Evaluate model on test samples---

100%|     | 4950/4950 [01:49<00:00, 45.06it/s]

---class aeroplane ap 0.5182232060568526---
---class bicycle ap 0.5928895331166626---
---class bird ap 0.457731500298588---
---class boat ap 0.303100847827348---
---class bottle ap 0.20191777763229413---
---class bus ap 0.61536422439128---
---class car ap 0.6348491083974395---
---class cat ap 0.6722168600532524---
---class chair ap 0.278135548608228---
---class cow ap 0.4256744403555651---
---class diningtable ap 0.3268612595417006---
---class dog ap 0.6281144088561152---
---class horse ap 0.6612629844143209---
---class motorbike ap 0.5394669509337908---
---class person ap 0.5128583117137296---
---class pottedplant ap 0.17264680245592376---
---class sheep ap 0.4384131127375066---
---class sofa ap 0.5061388734972234---
---class train ap 0.6908505299321648---
---class tvmonitor ap 0.4982419733310435---
---map 0.48374791270755146---
34 [0.5182232060568526, 0.5928895331166626, 0.457731500298588,
0.303100847827348, 0.20191777763229413, 0.61536422439128, 0.6348491083974395,
0.6722168600532524, 0.278135548608228, 0.4256744403555651, 0.3268612595417006,
0.6281144088561152, 0.6612629844143209, 0.5394669509337908, 0.5128583117137296,
0.17264680245592376, 0.4384131127375066, 0.5061388734972234, 0.6908505299321648,
0.4982419733310435]
Updating best test loss: 2.68101


Starting epoch 36 / 50
Learning Rate for this epoch: 0.0001
Epoch [36/50], Iter [50/209], Loss: total=1.842, reg=0.894,
containing_obj=0.566, no_obj=0.200, cls=0.181
Epoch [36/50], Iter [100/209], Loss: total=1.840, reg=0.895,
containing_obj=0.571, no_obj=0.200, cls=0.173
Epoch [36/50], Iter [150/209], Loss: total=1.863, reg=0.914,
containing_obj=0.576, no_obj=0.201, cls=0.172
Epoch [36/50], Iter [200/209], Loss: total=1.841, reg=0.897,
containing_obj=0.573, no_obj=0.201, cls=0.170


Starting epoch 37 / 50
```

```
Learning Rate for this epoch: 0.0001
Epoch [37/50], Iter [50/209], Loss: total=1.832, reg=0.890,
containing_obj=0.573, no_obj=0.199, cls=0.170
Epoch [37/50], Iter [100/209], Loss: total=1.839, reg=0.881,
containing_obj=0.579, no_obj=0.200, cls=0.179
Epoch [37/50], Iter [150/209], Loss: total=1.833, reg=0.873,
containing_obj=0.581, no_obj=0.201, cls=0.177
Epoch [37/50], Iter [200/209], Loss: total=1.836, reg=0.880,
containing_obj=0.576, no_obj=0.202, cls=0.178
Updating best test loss: 2.67865


Starting epoch 38 / 50
Learning Rate for this epoch: 0.0001
Epoch [38/50], Iter [50/209], Loss: total=1.809, reg=0.866,
containing_obj=0.564, no_obj=0.206, cls=0.172
Epoch [38/50], Iter [100/209], Loss: total=1.783, reg=0.849,
containing_obj=0.557, no_obj=0.203, cls=0.174
Epoch [38/50], Iter [150/209], Loss: total=1.811, reg=0.873,
containing_obj=0.562, no_obj=0.200, cls=0.176
Epoch [38/50], Iter [200/209], Loss: total=1.804, reg=0.873,
containing_obj=0.563, no_obj=0.200, cls=0.168


Starting epoch 39 / 50
Learning Rate for this epoch: 0.0001
Epoch [39/50], Iter [50/209], Loss: total=1.820, reg=0.875,
containing_obj=0.568, no_obj=0.197, cls=0.180
Epoch [39/50], Iter [100/209], Loss: total=1.804, reg=0.862,
containing_obj=0.569, no_obj=0.199, cls=0.173
Epoch [39/50], Iter [150/209], Loss: total=1.794, reg=0.854,
containing_obj=0.566, no_obj=0.202, cls=0.173
Epoch [39/50], Iter [200/209], Loss: total=1.807, reg=0.859,
containing_obj=0.569, no_obj=0.201, cls=0.178


Starting epoch 40 / 50
Learning Rate for this epoch: 0.0001
Epoch [40/50], Iter [50/209], Loss: total=1.796, reg=0.854,
containing_obj=0.565, no_obj=0.203, cls=0.174
Epoch [40/50], Iter [100/209], Loss: total=1.844, reg=0.885,
containing_obj=0.579, no_obj=0.203, cls=0.176
Epoch [40/50], Iter [150/209], Loss: total=1.809, reg=0.860,
containing_obj=0.571, no_obj=0.205, cls=0.173
Epoch [40/50], Iter [200/209], Loss: total=1.811, reg=0.862,
containing_obj=0.570, no_obj=0.206, cls=0.173
---Evaluate model on test samples---
```

```
100%|     | 4950/4950 [01:48<00:00, 45.50it/s]
---class aeroplane ap 0.4963578587792927---
---class bicycle ap 0.5756793539323094---
---class bird ap 0.4759911861761871---
---class boat ap 0.34074504406841394---
---class bottle ap 0.2197668206159354---
---class bus ap 0.6013150829459295---
---class car ap 0.6693489716075912---
---class cat ap 0.6831885360371106---
---class chair ap 0.2823448817599529---
---class cow ap 0.477146367867021---
---class diningtable ap 0.34567166923093595---
---class dog ap 0.6629475806430277---
---class horse ap 0.6844687741865834---
---class motorbike ap 0.5525202917014421---
---class person ap 0.521253363990281---
---class pottedplant ap 0.18267193056813083---
---class sheep ap 0.4471823407575993---
---class sofa ap 0.5170200961599739---
---class train ap 0.705163039335304---
---class tvmonitor ap 0.489696876633244---
---map 0.4965240033498134---
39 [0.4963578587792927, 0.5756793539323094, 0.4759911861761871,
0.34074504406841394, 0.2197668206159354, 0.6013150829459295, 0.6693489716075912,
0.6831885360371106, 0.2823448817599529, 0.477146367867021, 0.34567166923093595,
0.6629475806430277, 0.6844687741865834, 0.5525202917014421, 0.521253363990281,
0.18267193056813083, 0.4471823407575993, 0.5170200961599739, 0.705163039335304,
0.489696876633244]


Starting epoch 41 / 50
Learning Rate for this epoch: 1e-05
Epoch [41/50], Iter [50/209], Loss: total=1.853, reg=0.896,
containing_obj=0.565, no_obj=0.210, cls=0.181
Epoch [41/50], Iter [100/209], Loss: total=1.835, reg=0.871,
containing_obj=0.578, no_obj=0.206, cls=0.181
Epoch [41/50], Iter [150/209], Loss: total=1.830, reg=0.877,
containing_obj=0.574, no_obj=0.206, cls=0.173
Epoch [41/50], Iter [200/209], Loss: total=1.823, reg=0.876,
containing_obj=0.567, no_obj=0.206, cls=0.174
Updating best test loss: 2.67833


Starting epoch 42 / 50
Learning Rate for this epoch: 1e-05
Epoch [42/50], Iter [50/209], Loss: total=1.843, reg=0.896,
containing_obj=0.572, no_obj=0.203, cls=0.172
```

```
Epoch [42/50], Iter [100/209], Loss: total=1.780, reg=0.853,
containing_obj=0.554, no_obj=0.207, cls=0.167
Epoch [42/50], Iter [150/209], Loss: total=1.810, reg=0.868,
containing_obj=0.563, no_obj=0.204, cls=0.175
Epoch [42/50], Iter [200/209], Loss: total=1.799, reg=0.863,
containing_obj=0.563, no_obj=0.203, cls=0.170
Updating best test loss: 2.67742


Starting epoch 43 / 50
Learning Rate for this epoch: 1e-05
Epoch [43/50], Iter [50/209], Loss: total=1.832, reg=0.884,
containing_obj=0.576, no_obj=0.205, cls=0.167
Epoch [43/50], Iter [100/209], Loss: total=1.790, reg=0.849,
containing_obj=0.561, no_obj=0.210, cls=0.170
Epoch [43/50], Iter [150/209], Loss: total=1.801, reg=0.861,
containing_obj=0.567, no_obj=0.205, cls=0.167
Epoch [43/50], Iter [200/209], Loss: total=1.778, reg=0.845,
containing_obj=0.562, no_obj=0.204, cls=0.167


Starting epoch 44 / 50
Learning Rate for this epoch: 1e-05
Epoch [44/50], Iter [50/209], Loss: total=1.777, reg=0.857,
containing_obj=0.559, no_obj=0.204, cls=0.157
Epoch [44/50], Iter [100/209], Loss: total=1.758, reg=0.833,
containing_obj=0.560, no_obj=0.205, cls=0.160
Epoch [44/50], Iter [150/209], Loss: total=1.743, reg=0.823,
containing_obj=0.553, no_obj=0.203, cls=0.163
Epoch [44/50], Iter [200/209], Loss: total=1.774, reg=0.841,
containing_obj=0.563, no_obj=0.202, cls=0.168


Starting epoch 45 / 50
Learning Rate for this epoch: 1e-05
Epoch [45/50], Iter [50/209], Loss: total=1.715, reg=0.808,
containing_obj=0.533, no_obj=0.211, cls=0.162
Epoch [45/50], Iter [100/209], Loss: total=1.729, reg=0.816,
containing_obj=0.541, no_obj=0.209, cls=0.162
Epoch [45/50], Iter [150/209], Loss: total=1.779, reg=0.836,
containing_obj=0.566, no_obj=0.205, cls=0.172
Epoch [45/50], Iter [200/209], Loss: total=1.777, reg=0.839,
containing_obj=0.563, no_obj=0.206, cls=0.170
---Evaluate model on test samples---

100%|     | 4950/4950 [02:05<00:00, 39.43it/s]

---class aeroplane ap 0.512483127477515---
---class bicycle ap 0.5852869781267083---
```

```
---class bird ap 0.4803436960479395---
---class boat ap 0.32965797702728616---
---class bottle ap 0.230540525666382---
---class bus ap 0.6147311786261999---
---class car ap 0.6599836053944655---
---class cat ap 0.6758410380008759---
---class chair ap 0.2861696088756105---
---class cow ap 0.48210031349501703---
---class diningtable ap 0.352889673707156---
---class dog ap 0.6459057021764523---
---class horse ap 0.6800722319896104---
---class motorbike ap 0.5555304880513685---
---class person ap 0.5207299831311535---
---class pottedplant ap 0.1750900289035132---
---class sheep ap 0.45206182152757535---
---class sofa ap 0.505239520641471---
---class train ap 0.7003660216670626---
---class tvmonitor ap 0.5124488150235984---
---map 0.49787361677784797---
44 [0.512483127477515, 0.5852869781267083, 0.4803436960479395,
0.32965797702728616, 0.230540525666382, 0.6147311786261999, 0.6599836053944655,
0.6758410380008759, 0.2861696088756105, 0.48210031349501703, 0.352889673707156,
0.6459057021764523, 0.6800722319896104, 0.5555304880513685, 0.5207299831311535,
0.1750900289035132, 0.45206182152757535, 0.505239520641471, 0.7003660216670626,
0.5124488150235984]
Updating best test loss: 2.67660


Starting epoch 46 / 50
Learning Rate for this epoch: 1e-05
Epoch [46/50], Iter [50/209], Loss: total=1.811, reg=0.854,
containing_obj=0.567, no_obj=0.206, cls=0.184
Epoch [46/50], Iter [100/209], Loss: total=1.745, reg=0.826,
containing_obj=0.543, no_obj=0.205, cls=0.172
Epoch [46/50], Iter [150/209], Loss: total=1.768, reg=0.840,
containing_obj=0.553, no_obj=0.205, cls=0.170
Epoch [46/50], Iter [200/209], Loss: total=1.780, reg=0.848,
containing_obj=0.555, no_obj=0.205, cls=0.172


Starting epoch 47 / 50
Learning Rate for this epoch: 1e-05
Epoch [47/50], Iter [50/209], Loss: total=1.707, reg=0.805,
containing_obj=0.537, no_obj=0.204, cls=0.162
Epoch [47/50], Iter [100/209], Loss: total=1.759, reg=0.839,
containing_obj=0.548, no_obj=0.204, cls=0.168
Epoch [47/50], Iter [150/209], Loss: total=1.766, reg=0.840,
containing_obj=0.555, no_obj=0.204, cls=0.166
```

```
Epoch [47/50], Iter [200/209], Loss: total=1.777, reg=0.848,
containing_obj=0.560, no_obj=0.203, cls=0.167
Updating best test loss: 2.67653


Starting epoch 48 / 50
Learning Rate for this epoch: 1e-05
Epoch [48/50], Iter [50/209], Loss: total=1.817, reg=0.891,
containing_obj=0.555, no_obj=0.207, cls=0.164
Epoch [48/50], Iter [100/209], Loss: total=1.751, reg=0.842,
containing_obj=0.545, no_obj=0.205, cls=0.158
Epoch [48/50], Iter [150/209], Loss: total=1.755, reg=0.840,
containing_obj=0.551, no_obj=0.202, cls=0.162
Epoch [48/50], Iter [200/209], Loss: total=1.771, reg=0.847,
containing_obj=0.559, no_obj=0.204, cls=0.161


Starting epoch 49 / 50
Learning Rate for this epoch: 1e-05
Epoch [49/50], Iter [50/209], Loss: total=1.755, reg=0.832,
containing_obj=0.550, no_obj=0.209, cls=0.164
Epoch [49/50], Iter [100/209], Loss: total=1.805, reg=0.878,
containing_obj=0.558, no_obj=0.207, cls=0.163
Epoch [49/50], Iter [150/209], Loss: total=1.783, reg=0.852,
containing_obj=0.558, no_obj=0.207, cls=0.166
Epoch [49/50], Iter [200/209], Loss: total=1.796, reg=0.861,
containing_obj=0.558, no_obj=0.207, cls=0.169


Starting epoch 50 / 50
Learning Rate for this epoch: 1e-05
Epoch [50/50], Iter [50/209], Loss: total=1.731, reg=0.820,
containing_obj=0.549, no_obj=0.203, cls=0.159
Epoch [50/50], Iter [100/209], Loss: total=1.793, reg=0.858,
containing_obj=0.569, no_obj=0.204, cls=0.162
Epoch [50/50], Iter [150/209], Loss: total=1.781, reg=0.851,
containing_obj=0.565, no_obj=0.203, cls=0.162
Epoch [50/50], Iter [200/209], Loss: total=1.786, reg=0.853,
containing_obj=0.558, no_obj=0.204, cls=0.170
---Evaluate model on test samples---

100%|     | 4950/4950 [01:49<00:00, 45.04it/s]

---class aeroplane ap 0.5173585424561593---
---class bicycle ap 0.583714545511857---
---class bird ap 0.470908732081237---
---class boat ap 0.3220532102661491---
---class bottle ap 0.22284560678560017---
---class bus ap 0.6070306394455354---
```

```
---class car ap 0.6581933465351568---
---class cat ap 0.6884812056830767---
---class chair ap 0.2801820451820764---
---class cow ap 0.4824978902261956---
---class diningtable ap 0.33360376561974125---
---class dog ap 0.659913921396504---
---class horse ap 0.693596605213306---
---class motorbike ap 0.5690485472680742---
---class person ap 0.5323960906993205---
---class pottedplant ap 0.18505575579428019---
---class sheep ap 0.44878501670270876---
---class sofa ap 0.5058467567143687---
---class train ap 0.6846409861475697---
---class tvmonitor ap 0.520872493525345---
---map 0.49835128516271326---
49 [0.5173585424561593, 0.583714545511857, 0.470908732081237,
0.3220532102661491, 0.22284560678560017, 0.6070306394455354, 0.6581933465351568,
0.6884812056830767, 0.2801820451820764, 0.4824978902261956, 0.33360376561974125,
0.659913921396504, 0.693596605213306, 0.5690485472680742, 0.5323960906993205,
0.18505575579428019, 0.44878501670270876, 0.5058467567143687,
0.6846409861475697, 0.520872493525345]
```

[ ]:
```python
load_network_path = 'checkpoints/best_detector.pth'
pretrained = True

# use to load a previously trained network
if load_network_path is not None:
    print('Loading saved network from {}'.format(load_network_path))
    net = resnet50().to(device)
    net.load_state_dict(torch.load(load_network_path))
else:
    print('Load pre-trained model')
    net = resnet50(pretrained=pretrained).to(device)
```

```
Loading saved network from checkpoints/best_detector.pth
```

[ ]:
```python
# train for another 20 epochs
best_test_loss = np.inf
learning_rate = 1e-7
num_epochs = 20
for epoch in range(num_epochs):
    net.train()

    # Update learning rate late in training
    # if epoch == 10:
    #     learning_rate /= 10.0
```

```python
    for param_group in optimizer.param_groups:
        param_group['lr'] = learning_rate

    print('\n\nStarting epoch %d / %d' % (epoch + 1, num_epochs))
    print('Learning Rate for this epoch: {}'.format(learning_rate))

    total_loss = collections.defaultdict(int)

    for i, data in enumerate(train_loader):
        data = (item.to(device) for item in data)
        images, target_boxes, target_cls, has_object_map = data
        pred = net(images)
        loss_dict = criterion(pred, target_boxes, target_cls, has_object_map)
        for key in loss_dict:
            total_loss[key] += loss_dict[key].item()

        optimizer.zero_grad()
        loss_dict['total_loss'].backward()
        optimizer.step()

        if (i+1) % 50 == 0:
            outstring = 'Epoch [%d/%d], Iter [%d/%d], Loss: ' % ((epoch+1,␣
↪num_epochs, i+1, len(train_loader)))
            outstring += ', '.join( "%s=%.3f" % (key[:-5], val / (i+1)) for␣
↪key, val in total_loss.items() )
            print(outstring)

    # evaluate the network on the test data
    if (epoch + 1) % 5 == 0:
        test_aps = evaluate(net, test_dataset_file=annotation_file_test,␣
↪img_root=file_root_test)
        print(epoch, test_aps)
    with torch.no_grad():
        test_loss = 0.0
        net.eval()
        for i, data in enumerate(test_loader):
            data = (item.to(device) for item in data)
            images, target_boxes, target_cls, has_object_map = data

            pred = net(images)
            loss_dict = criterion(pred, target_boxes, target_cls,␣
↪has_object_map)
            test_loss += loss_dict['total_loss'].item()
        test_loss /= len(test_loader)

    if best_test_loss > test_loss:
        best_test_loss = test_loss
```

```python
        print('Updating best test loss: %.5f' % best_test_loss)
        # torch.save(net.state_dict(),'checkpoints/best_detector.pth')

    if (epoch+1) in [5, 10]:
        torch.save(net.state_dict(),'checkpoints/detector_epoch_%d.pth' %␣
↪(epoch+1))

    torch.save(net.state_dict(),'checkpoints/detector.pth')
```

```
Starting epoch 1 / 20
Learning Rate for this epoch: 1e-07
Epoch [1/20], Iter [50/105], Loss: total=1.752, reg=0.847, containing_obj=0.559,
no_obj=0.202, cls=0.144
Epoch [1/20], Iter [100/105], Loss: total=1.760, reg=0.856,
containing_obj=0.555, no_obj=0.201, cls=0.148
Updating best test loss: 2.67042


Starting epoch 2 / 20
Learning Rate for this epoch: 1e-07
Epoch [2/20], Iter [50/105], Loss: total=1.727, reg=0.813, containing_obj=0.552,
no_obj=0.205, cls=0.157
Epoch [2/20], Iter [100/105], Loss: total=1.738, reg=0.829,
containing_obj=0.554, no_obj=0.205, cls=0.150
Updating best test loss: 2.66498


Starting epoch 3 / 20
Learning Rate for this epoch: 1e-07
Epoch [3/20], Iter [50/105], Loss: total=1.729, reg=0.835, containing_obj=0.545,
no_obj=0.202, cls=0.148
Epoch [3/20], Iter [100/105], Loss: total=1.777, reg=0.857,
containing_obj=0.559, no_obj=0.202, cls=0.159


Starting epoch 4 / 20
Learning Rate for this epoch: 1e-07
Epoch [4/20], Iter [50/105], Loss: total=1.817, reg=0.869, containing_obj=0.584,
no_obj=0.201, cls=0.162
Epoch [4/20], Iter [100/105], Loss: total=1.756, reg=0.843,
containing_obj=0.557, no_obj=0.202, cls=0.153
Updating best test loss: 2.66042


Starting epoch 5 / 20
```

```
Learning Rate for this epoch: 1e-07
Epoch [5/20], Iter [50/105], Loss: total=1.751, reg=0.833, containing_obj=0.564,
no_obj=0.203, cls=0.149
Epoch [5/20], Iter [100/105], Loss: total=1.756, reg=0.840,
containing_obj=0.562, no_obj=0.203, cls=0.152
---Evaluate model on test samples---

100%|     | 4950/4950 [01:45<00:00, 46.90it/s]

---class aeroplane ap 0.5156360272465859---
---class bicycle ap 0.5821791933248942---
---class bird ap 0.4754608255767426---
---class boat ap 0.32956161643160997---
---class bottle ap 0.23962633438956316---
---class bus ap 0.6098380274170774---
---class car ap 0.6648527938897819---
---class cat ap 0.6839515576802225---
---class chair ap 0.2838002041505672---
---class cow ap 0.47104410313192613---
---class diningtable ap 0.3477910782984703---
---class dog ap 0.6620648664117479---
---class horse ap 0.6918500943034918---
---class motorbike ap 0.5617132807435516---
---class person ap 0.5268194852696351---
---class pottedplant ap 0.17710022135345396---
---class sheep ap 0.46297048589591366---
---class sofa ap 0.5044173320168852---
---class train ap 0.7046449198400089---
---class tvmonitor ap 0.507086273890087---
---map 0.5001204360631107---
4 [0.5156360272465859, 0.5821791933248942, 0.4754608255767426,
0.32956161643160997, 0.23962633438956316, 0.6098380274170774,
0.6648527938897819, 0.6839515576802225, 0.2838002041505672, 0.47104410313192613,
0.3477910782984703, 0.6620648664117479, 0.6918500943034918, 0.5617132807435516,
0.5268194852696351, 0.17710022135345396, 0.46297048589591366,
0.5044173320168852, 0.7046449198400089, 0.507086273890087]
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Cell In[16], line 45
     43 net.eval()
     44 for i, data in enumerate(test_loader):
---> 45     data = (item.to(device) for item in data)
     46     images, target_boxes, target_cls, has_object_map = data
     48     pred = net(images)

KeyboardInterrupt:
```

# 1 View example predictions

[ ]:
```
net.eval()

# select random image from test set
image_name = random.choice(test_dataset.fnames)
image = cv2.imread(os.path.join(file_root_test, image_name))
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

print('predicting...')
result = predict_image(net, image_name, root_img_directory=file_root_test)
for left_up, right_bottom, class_name, _, prob in result:
    color = COLORS[VOC_CLASSES.index(class_name)]
    cv2.rectangle(image, left_up, right_bottom, color, 2)
    label = class_name + str(round(prob, 2))
    text_size, baseline = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.4,␣
 ↪1)
    p1 = (left_up[0], left_up[1] - text_size[1])
    cv2.rectangle(image, (p1[0] - 2 // 2, p1[1] - 2 - baseline), (p1[0] +␣
 ↪text_size[0], p1[1] + text_size[1]),
                  color, -1)
    cv2.putText(image, label, (p1[0], p1[1] + baseline), cv2.
 ↪FONT_HERSHEY_SIMPLEX, 0.4, (255, 255, 255), 1, 8)

plt.figure(figsize = (15,15))
plt.imshow(image)
```

predicting…

[ ]: <matplotlib.image.AxesImage at 0x14dc30942f50>


## 1.1 Evaluate on Test

To evaluate detection results we use mAP (mean of average precision over each class)

```
[ ]: test_aps = evaluate(net, test_dataset_file=annotation_file_test,␣
     ↪img_root=file_root_test)
```

```
---Evaluate model on test samples---

100%|      | 4950/4950 [01:44<00:00, 47.40it/s]

---class aeroplane ap 0.5156360272465859---
---class bicycle ap 0.5821791933248942---
---class bird ap 0.4754608255767426---
---class boat ap 0.32956161643160997---
---class bottle ap 0.23962633438956316---
---class bus ap 0.6098380274170774---
---class car ap 0.6648527938897819---
---class cat ap 0.6839515576802225---
---class chair ap 0.2838002041505672---
---class cow ap 0.47104410313192613---
---class diningtable ap 0.3477910782984703---
```

```
---class dog ap 0.6620648664117479---
---class horse ap 0.6918500943034918---
---class motorbike ap 0.5617132807435516---
---class person ap 0.5268194852696351---
---class pottedplant ap 0.17710022135345396---
---class sheep ap 0.46297048589591366---
---class sofa ap 0.5044173320168852---
---class train ap 0.7046449198400089---
---class tvmonitor ap 0.507086273890087---
---map 0.5001204360631107---
```

### 1.1.1 Cell added to get intermediate mAP values for students

```
[ ]: network_paths = ['./checkpoints/detector_epoch_%d.pth' % epoch for epoch in [5,␣
     ↪10, 20, 30, 40]]+['./checkpoints/detector.pth']
     for load_network_path in network_paths:
         print('Loading saved network from {}'.format(load_network_path))
         net_loaded =  resnet50().to(device)
         net_loaded.load_state_dict(torch.load(load_network_path))
         evaluate(net_loaded, test_dataset_file=annotation_file_test,␣
     ↪img_root=file_root_test)
```

```
Loading saved network from ./checkpoints/detector_epoch_5.pth
---Evaluate model on test samples---

100%|     | 4950/4950 [01:33<00:00, 53.02it/s]

---class aeroplane ap 0.07658872725054766---
---class bicycle ap 0.05062845689428884---
---class bird ap 0.06031496703720878---
---class boat ap 0.015634869227697974---
---class bottle ap 0.0161706925891787---
---class bus ap 0.0--- (no predictions for this class)
---class car ap 0.12292450106729097---
---class cat ap 0.0111731843575419---
---class chair ap 0.03533659231025792---
---class cow ap 0.0---
---class diningtable ap 0.0--- (no predictions for this class)
---class dog ap 0.004703476482617587---
---class horse ap 0.05198014629049112---
---class motorbike ap 0.0--- (no predictions for this class)
---class person ap 0.04971351804509193---
---class pottedplant ap 0.009684292097146533---
---class sheep ap 0.019559228650137744---
---class sofa ap 0.0--- (no predictions for this class)
---class train ap 0.0--- (no predictions for this class)
---class tvmonitor ap 0.16219807005625944---
---map 0.03433053611778785---
Loading saved network from ./checkpoints/detector_epoch_10.pth
```

```
---Evaluate model on test samples---

100%|     | 4950/4950 [01:41<00:00, 48.83it/s]

---class aeroplane ap 0.29672594414934905---
---class bicycle ap 0.3422849102545965---
---class bird ap 0.26962383409735075---
---class boat ap 0.11380292495688714---
---class bottle ap 0.0465665892740073---
---class bus ap 0.25234830916967904---
---class car ap 0.4968924330943932---
---class cat ap 0.5211356780331985---
---class chair ap 0.16107418955618402---
---class cow ap 0.14420768319136462---
---class diningtable ap 0.0048543689320388345---
---class dog ap 0.44054838096319154---
---class horse ap 0.44923080267119464---
---class motorbike ap 0.3760258183215436---
---class person ap 0.2922193643238318---
---class pottedplant ap 0.06539052004378372---
---class sheep ap 0.18705394618455345---
---class sofa ap 0.16438100439114636---
---class train ap 0.45297452404630134---
---class tvmonitor ap 0.3515317055710824---
---map 0.2714436465612839---
Loading saved network from ./checkpoints/detector_epoch_20.pth
---Evaluate model on test samples---

100%|     | 4950/4950 [01:51<00:00, 44.27it/s]

---class aeroplane ap 0.44728433469512224---
---class bicycle ap 0.5209844377553562---
---class bird ap 0.4001112153420066---
---class boat ap 0.1782360244094975---
---class bottle ap 0.12431433578342119---
---class bus ap 0.49089998788611183---
---class car ap 0.5543484487470753---
---class cat ap 0.5593981607331366---
---class chair ap 0.22098080775915271---
---class cow ap 0.42783670584498845---
---class diningtable ap 0.17728054829978707---
---class dog ap 0.5597769708665804---
---class horse ap 0.6254283583663549---
---class motorbike ap 0.5664770219279924---
---class person ap 0.4564772852679984---
---class pottedplant ap 0.1360019135184708---
---class sheep ap 0.3841037883120979---
---class sofa ap 0.4167732522132447---
---class train ap 0.5855054093211073---
---class tvmonitor ap 0.3582455920919174---
```

```
---map 0.40952322995707097---
Loading saved network from ./checkpoints/detector_epoch_30.pth
---Evaluate model on test samples---

100%|     | 4950/4950 [01:51<00:00, 44.30it/s]

---class aeroplane ap 0.4719819809800481---
---class bicycle ap 0.5780565318334638---
---class bird ap 0.4494709722559183---
---class boat ap 0.2924411458666887---
---class bottle ap 0.17512803769885577---
---class bus ap 0.562238838562554---
---class car ap 0.6349132255562765---
---class cat ap 0.6546596536208156---
---class chair ap 0.2358103752066274---
---class cow ap 0.34905603398848234---
---class diningtable ap 0.2802260169057064---
---class dog ap 0.5733626063495525---
---class horse ap 0.6389986491725825---
---class motorbike ap 0.5315041878862564---
---class person ap 0.49390194785564623---
---class pottedplant ap 0.15866587891490264---
---class sheep ap 0.4187775294345249---
---class sofa ap 0.42169926734926033---
---class train ap 0.63268485842851---
---class tvmonitor ap 0.47652801270692446---
---map 0.45150528752867986---
Loading saved network from ./checkpoints/detector_epoch_40.pth
---Evaluate model on test samples---

100%|     | 4950/4950 [01:48<00:00, 45.72it/s]

---class aeroplane ap 0.4963578587792927---
---class bicycle ap 0.5756793539323094---
---class bird ap 0.4759911861761871---
---class boat ap 0.34074504406841394---
---class bottle ap 0.2197668206159354---
---class bus ap 0.6013150829459295---
---class car ap 0.6693489716075912---
---class cat ap 0.6831885360371106---
---class chair ap 0.2823448817599529---
---class cow ap 0.477146367867021---
---class diningtable ap 0.34567166923093595---
---class dog ap 0.6629475806430277---
---class horse ap 0.6844687741865834---
---class motorbike ap 0.5525202917014421---
---class person ap 0.521253363990281---
---class pottedplant ap 0.18267193056813083---
---class sheep ap 0.4471823407575993---
---class sofa ap 0.5170200961599739---
```

```
---class train ap 0.705163039335304---
---class tvmonitor ap 0.489696876633244---
---map 0.4965240033498134---
Loading saved network from ./checkpoints/detector.pth
---Evaluate model on test samples---

100%|      | 4950/4950 [02:00<00:00, 40.91it/s]

---class aeroplane ap 0.5173585424561593---
---class bicycle ap 0.583714545511857---
---class bird ap 0.470908732081237---
---class boat ap 0.3220532102661491---
---class bottle ap 0.22284560678560017---
---class bus ap 0.6070306394455354---
---class car ap 0.6581933465351568---
---class cat ap 0.6884812056830767---
---class chair ap 0.2801820451820764---
---class cow ap 0.4824978902261956---
---class diningtable ap 0.33360376561974125---
---class dog ap 0.659913921396504---
---class horse ap 0.693596605213306---
---class motorbike ap 0.5690485472680742---
---class person ap 0.5323960906993205---
---class pottedplant ap 0.18505575579428019---
---class sheep ap 0.44878501670270876---
---class sofa ap 0.5058467567143687---
---class train ap 0.6846409861475697---
---class tvmonitor ap 0.520872493525345---
---map 0.49835128516271326---
```

[ ]:
```python
output_submission_csv('my_new_solution.csv', test_aps)
```

## 2 Extra Credit 1: Video Object Detection

[ ]:
```python
import numpy as np
import cv2
import pafy
import matplotlib.pyplot as plt
from matplotlib import cm
from PIL import Image
from tqdm import trange

import torch
from torch import nn
from torchvision import transforms

url = "https://www.youtube.com/watch?v=xZGahvrep3o"
```

[ ]:
```
!pip install pytube
```

```
Collecting pytube
  Downloading pytube-15.0.0-py3-none-any.whl.metadata (5.0 kB)
Downloading pytube-15.0.0-py3-none-any.whl (57 kB)
                         57.6/57.6 kB
297.8 kB/s eta 0:00:000:00:01
Installing collected packages: pytube
Successfully installed pytube-15.0.0
```

[ ]:
```
### Download YouTube Video ###
from pytube import YouTube
url = "https://www.youtube.com/watch?v=xZGahvrep3o"
path = "./video"
video = YouTube(url)
video.streams.filter(progressive=True)
stream = video.streams.get_by_itag(22)
stream.download(path)
```

[ ]: '/u/qilong/mistral/mistral2/./video/SNL Digital Short YOLO - SNL.mp4'

[ ]:
```
### Video to Frame ###

def video2frame(videos_path,frames_save_path,time_interval):

    '''
    :param videos_path: path to video
    :param frames_save_path: path to frames
    :param time_interval: time interval
    :return:
    '''
    vidcap = cv2.VideoCapture(videos_path)
    success, image = vidcap.read()
    count = 0
    while success:
        success, image = vidcap.read()
        count += 1
        if count % time_interval == 0:
            try:
                cv2.imencode('.jpg', image)[1].tofile(frames_save_path + "/%d.
 ↪jpg" % count)
            except:
                break
    print(count)

    return count
```

[ ]:
```
videos_path = r'./video/SNL_Digital_Short_YOLO_SNL.mp4'
frames_save_path = r'./video/frame2'
time_interval = 2
count = video2frame(videos_path, frames_save_path, time_interval)
```

5265

[ ]:
```
### Load Detector ###
load_network_path = "checkpoints/best_detector.pth"
pretrained = True

# use to load a previously trained network
if load_network_path is not None:
    print('Loading saved network from {}'.format(load_network_path))
    net = resnet50().to(device)
    net.load_state_dict(torch.load(load_network_path))
else:
    print('Load pre-trained model')
    net = resnet50(pretrained=pretrained).to(device)
```

Loading saved network from checkpoints/best_detector.pth

[ ]:
```
### Test on one frame ###
net.eval()
detected_save_path = "./video/detected2"

# load image from frame
idx = 100
image = cv2.imread(frames_save_path + "/{}.jpg".format(idx * 10))
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

result = predict_image(net, "{}.jpg".format(idx * 10),␣
↪root_img_directory=frames_save_path + "/")
for left_up, right_bottom, class_name, _, prob in result:
    color = COLORS[VOC_CLASSES.index(class_name)]
    cv2.rectangle(image, left_up, right_bottom, color, 2)
    label = class_name + str(round(prob, 2))
    text_size, baseline = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.4,␣
↪1)
    p1 = (left_up[0], left_up[1] - text_size[1])
    cv2.rectangle(image, (p1[0] - 2 // 2, p1[1] - 2 - baseline), (p1[0] +␣
↪text_size[0], p1[1] + text_size[1]),
                  color, -1)
    cv2.putText(image, label, (p1[0], p1[1] + baseline), cv2.
↪FONT_HERSHEY_SIMPLEX, 0.4, (255, 255, 255), 1, 8)

# show image
```

```
plt.figure(figsize = (15,15))
plt.imshow(image)

# save image
cv2.imwrite(detected_save_path + "/{}.jpg".format(idx * 10), image)
```

[ ]: True


```
[ ]: ### Detection ###
net.eval()
detected_save_path = "./video/detected2"
if count is None:
    count = 5265

# load image from frame
for i in trange(count // time_interval):
    idx = i + 1
    try:
        image = cv2.imread("./video/frame2/{}.jpg".format(idx * time_interval))
    except:
        print("Can not load frame {}".format(idx * time_interval))
        continue
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    result = predict_image(net, "{}.jpg".format(idx * time_interval),␣
 ↪root_img_directory="./video/frame2/")
```

```
    for left_up, right_bottom, class_name, _, prob in result:
        color = COLORS[VOC_CLASSES.index(class_name)]
        cv2.rectangle(image, left_up, right_bottom, color, 2)
        label = class_name + str(round(prob, 2))
        text_size, baseline = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX,␣
↪0.4, 1)
        p1 = (left_up[0], left_up[1] - text_size[1])
        cv2.rectangle(image, (p1[0] - 2 // 2, p1[1] - 2 - baseline), (p1[0] +␣
↪text_size[0], p1[1] + text_size[1]), color, -1)
        cv2.putText(image, label, (p1[0], p1[1] + baseline), cv2.
↪FONT_HERSHEY_SIMPLEX, 0.4, (255, 255, 255), 1, 8)

    # save image
    cv2.imwrite(detected_save_path + "/{}.jpg".format(idx * time_interval),␣
↪image)
```

```
100%|      | 2632/2632 [02:17<00:00, 19.16it/s]
```

[ ]:

```
### Frame to Video ###

def frame2video(im_dir,video_dir,fps):

    im_list = []
    for idx in range(1, count // time_interval + 1):
        im_list.append("{}.jpg".format(idx * time_interval))
    img = Image.open(im_dir + im_list[100])
    img_size = img.size
    print(img_size)

    fourcc = cv2.VideoWriter_fourcc(*'mp4v')
    videoWriter = cv2.VideoWriter(video_dir, fourcc, fps, img_size)

    print("Start producing video...")
    for i in trange(len(im_list)):
        im_name = im_dir + im_list[i]
        frame = cv2.imread(im_name)
        videoWriter.write(frame)
    videoWriter.release()
    print('finish')
```

[ ]:

```
im_dir =  detected_save_path + "/"
video_dir = r'./video/detected2_video.mp4'
fps = 24
frame2video(im_dir, video_dir, fps)
```

```
(1280, 720)
Start producing video…
```

```
100%|      | 2632/2632 [00:32<00:00, 81.15it/s]

finish
```

## 3 Extra Credit 2: Better Pretrained Model

[ ]:
```python
from src.resnet_yolo import resnet101

load_network_path = None #'checkpoints/best_detector.pth'
pretrained = True

# use to load a previously trained network
if load_network_path is not None:
    print('Loading saved network from {}'.format(load_network_path))
    net = resnet101().to(device)
    net.load_state_dict(torch.load(load_network_path))
else:
    print('Load pre-trained model')
    net = resnet101(pretrained=pretrained).to(device)
```

```
Load pre-trained model
```

[ ]:
```python
learning_rate = 0.001
num_epochs = 50
batch_size = 24

# Yolo loss component coefficients (as given in Yolo v1 paper)
lambda_coord = 5
lambda_noobj = 0.5

criterion = YoloLoss(S, B, lambda_coord, lambda_noobj)
optimizer = torch.optim.SGD(net.parameters(), lr=learning_rate, momentum=0.9,␣
 ↪weight_decay=5e-4)
```

[ ]:
```python
best_test_loss = np.inf
for epoch in range(num_epochs):
    net.train()

    # Update learning rate late in training
    if epoch == 30 or epoch == 40:
        learning_rate /= 10.0

    for param_group in optimizer.param_groups:
        param_group['lr'] = learning_rate

    print('\n\nStarting epoch %d / %d' % (epoch + 1, num_epochs))
```

```
    print('Learning Rate for this epoch: {}'.format(learning_rate))

    total_loss = collections.defaultdict(int)

    for i, data in enumerate(train_loader):
        data = (item.to(device) for item in data)
        images, target_boxes, target_cls, has_object_map = data
        pred = net(images)
        loss_dict = criterion(pred, target_boxes, target_cls, has_object_map)
        for key in loss_dict:
            total_loss[key] += loss_dict[key].item()

        optimizer.zero_grad()
        loss_dict['total_loss'].backward()
        optimizer.step()

        if (i+1) % 50 == 0:
            outstring = 'Epoch [%d/%d], Iter [%d/%d], Loss: ' % ((epoch+1, num_epochs, i+1, len(train_loader)))
            outstring += ', '.join( "%s=%.3f" % (key[:-5], val / (i+1)) for key, val in total_loss.items() )
            print(outstring)

    # evaluate the network on the test data
    if (epoch + 1) % 5 == 0:
        test_aps = evaluate(net, test_dataset_file=annotation_file_test, img_root=file_root_test)
        print(epoch, test_aps)
    with torch.no_grad():
        test_loss = 0.0
        net.eval()
        for i, data in enumerate(test_loader):
            data = (item.to(device) for item in data)
            images, target_boxes, target_cls, has_object_map = data

            pred = net(images)
            loss_dict = criterion(pred, target_boxes, target_cls, has_object_map)
            test_loss += loss_dict['total_loss'].item()
        test_loss /= len(test_loader)

    if best_test_loss > test_loss:
        best_test_loss = test_loss
        print('Updating best test loss: %.5f' % best_test_loss)
        torch.save(net.state_dict(),'checkpoints/res101_best_detector.pth')

    if (epoch+1) in [5, 10, 20, 30, 40]:
```

```
        torch.save(net.state_dict(),'checkpoints/res101_detector_epoch_%d.pth'␣
↪% (epoch+1))

    torch.save(net.state_dict(),'checkpoints/res101_detector.pth')
```

```
Starting epoch 1 / 50
Learning Rate for this epoch: 0.001
Epoch [1/50], Iter [50/209], Loss: total=23.906, reg=3.820,
containing_obj=0.347, no_obj=12.476, cls=7.263
Epoch [1/50], Iter [100/209], Loss: total=15.517, reg=3.252,
containing_obj=0.453, no_obj=6.467, cls=5.345
Epoch [1/50], Iter [150/209], Loss: total=12.311, reg=2.919,
containing_obj=0.516, no_obj=4.418, cls=4.458
Epoch [1/50], Iter [200/209], Loss: total=10.481, reg=2.697,
containing_obj=0.555, no_obj=3.377, cls=3.852
Updating best test loss: 5.02656


Starting epoch 2 / 50
Learning Rate for this epoch: 0.001
Epoch [2/50], Iter [50/209], Loss: total=4.768, reg=1.978, containing_obj=0.740,
no_obj=0.204, cls=1.846
Epoch [2/50], Iter [100/209], Loss: total=4.751, reg=1.962,
containing_obj=0.764, no_obj=0.190, cls=1.835
Epoch [2/50], Iter [150/209], Loss: total=4.743, reg=1.963,
containing_obj=0.778, no_obj=0.179, cls=1.824
Epoch [2/50], Iter [200/209], Loss: total=4.652, reg=1.935,
containing_obj=0.786, no_obj=0.169, cls=1.761
Updating best test loss: 4.39795


Starting epoch 3 / 50
Learning Rate for this epoch: 0.001
Epoch [3/50], Iter [50/209], Loss: total=4.397, reg=1.890, containing_obj=0.864,
no_obj=0.126, cls=1.517
Epoch [3/50], Iter [100/209], Loss: total=4.289, reg=1.848,
containing_obj=0.862, no_obj=0.121, cls=1.457
Epoch [3/50], Iter [150/209], Loss: total=4.225, reg=1.826,
containing_obj=0.870, no_obj=0.117, cls=1.413
Epoch [3/50], Iter [200/209], Loss: total=4.187, reg=1.821,
containing_obj=0.872, no_obj=0.113, cls=1.381
Updating best test loss: 4.08262


Starting epoch 4 / 50
```

```
Learning Rate for this epoch: 0.001
Epoch [4/50], Iter [50/209], Loss: total=3.852, reg=1.713, containing_obj=0.898,
no_obj=0.094, cls=1.146
Epoch [4/50], Iter [100/209], Loss: total=3.823, reg=1.693,
containing_obj=0.903, no_obj=0.092, cls=1.135
Epoch [4/50], Iter [150/209], Loss: total=3.813, reg=1.702,
containing_obj=0.908, no_obj=0.089, cls=1.114
Epoch [4/50], Iter [200/209], Loss: total=3.791, reg=1.705,
containing_obj=0.912, no_obj=0.087, cls=1.087
Updating best test loss: 3.77741


Starting epoch 5 / 50
Learning Rate for this epoch: 0.001
Epoch [5/50], Iter [50/209], Loss: total=3.573, reg=1.631, containing_obj=0.915,
no_obj=0.077, cls=0.951
Epoch [5/50], Iter [100/209], Loss: total=3.568, reg=1.636,
containing_obj=0.925, no_obj=0.075, cls=0.931
Epoch [5/50], Iter [150/209], Loss: total=3.542, reg=1.627,
containing_obj=0.928, no_obj=0.074, cls=0.913
Epoch [5/50], Iter [200/209], Loss: total=3.558, reg=1.639,
containing_obj=0.937, no_obj=0.073, cls=0.909
---Evaluate model on test samples---

100%|     | 4950/4950 [02:15<00:00, 36.62it/s]

---class aeroplane ap 0.0---
---class bicycle ap 0.0--- (no predictions for this class)
---class bird ap 0.0--- (no predictions for this class)
---class boat ap 0.0--- (no predictions for this class)
---class bottle ap 0.0--- (no predictions for this class)
---class bus ap 0.0--- (no predictions for this class)
---class car ap 0.0--- (no predictions for this class)
---class cat ap 0.0--- (no predictions for this class)
---class chair ap 0.0--- (no predictions for this class)
---class cow ap 0.0--- (no predictions for this class)
---class diningtable ap 0.0--- (no predictions for this class)
---class dog ap 0.0--- (no predictions for this class)
---class horse ap 0.0--- (no predictions for this class)
---class motorbike ap 0.0--- (no predictions for this class)
---class person ap 0.0--- (no predictions for this class)
---class pottedplant ap 0.0--- (no predictions for this class)
---class sheep ap 0.0--- (no predictions for this class)
---class sofa ap 0.0--- (no predictions for this class)
---class train ap 0.0--- (no predictions for this class)
---class tvmonitor ap 0.0--- (no predictions for this class)
---map 0.0---
4 [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0]
```

```
Updating best test loss: 3.66728


Starting epoch 6 / 50
Learning Rate for this epoch: 0.001
Epoch [6/50], Iter [50/209], Loss: total=3.335, reg=1.534, containing_obj=0.952,
no_obj=0.066, cls=0.784
Epoch [6/50], Iter [100/209], Loss: total=3.348, reg=1.557,
containing_obj=0.945, no_obj=0.065, cls=0.780
Epoch [6/50], Iter [150/209], Loss: total=3.405, reg=1.583,
containing_obj=0.971, no_obj=0.064, cls=0.787
Epoch [6/50], Iter [200/209], Loss: total=3.371, reg=1.571,
containing_obj=0.966, no_obj=0.063, cls=0.771
Updating best test loss: 3.58269


Starting epoch 7 / 50
Learning Rate for this epoch: 0.001
Epoch [7/50], Iter [50/209], Loss: total=3.310, reg=1.555, containing_obj=0.997,
no_obj=0.059, cls=0.699
Epoch [7/50], Iter [100/209], Loss: total=3.281, reg=1.546,
containing_obj=0.994, no_obj=0.059, cls=0.682
Epoch [7/50], Iter [150/209], Loss: total=3.210, reg=1.513,
containing_obj=0.979, no_obj=0.058, cls=0.660
Epoch [7/50], Iter [200/209], Loss: total=3.220, reg=1.514,
containing_obj=0.985, no_obj=0.058, cls=0.663
Updating best test loss: 3.50733


Starting epoch 8 / 50
Learning Rate for this epoch: 0.001
Epoch [8/50], Iter [50/209], Loss: total=3.192, reg=1.520, containing_obj=1.012,
no_obj=0.056, cls=0.604
Epoch [8/50], Iter [100/209], Loss: total=3.126, reg=1.492,
containing_obj=0.992, no_obj=0.056, cls=0.587
Epoch [8/50], Iter [150/209], Loss: total=3.119, reg=1.486,
containing_obj=0.992, no_obj=0.056, cls=0.585
Epoch [8/50], Iter [200/209], Loss: total=3.110, reg=1.481,
containing_obj=0.989, no_obj=0.056, cls=0.584
Updating best test loss: 3.36429


Starting epoch 9 / 50
Learning Rate for this epoch: 0.001
Epoch [9/50], Iter [50/209], Loss: total=3.065, reg=1.489, containing_obj=0.973,
no_obj=0.060, cls=0.543
```

```
Epoch [9/50], Iter [100/209], Loss: total=3.048, reg=1.460,
containing_obj=0.972, no_obj=0.062, cls=0.553
Epoch [9/50], Iter [150/209], Loss: total=3.024, reg=1.453,
containing_obj=0.956, no_obj=0.067, cls=0.548
Epoch [9/50], Iter [200/209], Loss: total=2.995, reg=1.442,
containing_obj=0.942, no_obj=0.074, cls=0.537
Updating best test loss: 3.21619


Starting epoch 10 / 50
Learning Rate for this epoch: 0.001
Epoch [10/50], Iter [50/209], Loss: total=2.815, reg=1.380,
containing_obj=0.843, no_obj=0.127, cls=0.465
Epoch [10/50], Iter [100/209], Loss: total=2.853, reg=1.405,
containing_obj=0.843, no_obj=0.132, cls=0.473
Epoch [10/50], Iter [150/209], Loss: total=2.828, reg=1.380,
containing_obj=0.845, no_obj=0.134, cls=0.469
Epoch [10/50], Iter [200/209], Loss: total=2.833, reg=1.388,
containing_obj=0.837, no_obj=0.135, cls=0.473
---Evaluate model on test samples---

100%|     | 4950/4950 [02:16<00:00, 36.20it/s]

---class aeroplane ap 0.37014882268865834---
---class bicycle ap 0.33202984858019907---
---class bird ap 0.32980370752002586---
---class boat ap 0.17872398942594217---
---class bottle ap 0.03817529058860585---
---class bus ap 0.39699784808851935---
---class car ap 0.5011312021898204---
---class cat ap 0.48249772972528926---
---class chair ap 0.166253515159359---
---class cow ap 0.3905101878308053---
---class diningtable ap 0.0048543689320388345---
---class dog ap 0.5036399047043938---
---class horse ap 0.5081073248781507---
---class motorbike ap 0.20567201037109883---
---class person ap 0.37773878026190766---
---class pottedplant ap 0.0719055271847999---
---class sheep ap 0.21640730222603838---
---class sofa ap 0.14106169740623525---
---class train ap 0.49811441302951087---
---class tvmonitor ap 0.40449863737274727---
---map 0.3059136054082073---
9 [0.37014882268865834, 0.33202984858019907, 0.32980370752002586,
0.17872398942594217, 0.03817529058860585, 0.39699784808851935,
0.5011312021898204, 0.48249772972528926, 0.166253515159359, 0.3905101878308053,
0.0048543689320388345, 0.5036399047043938, 0.5081073248781507,
0.20567201037109883, 0.37773878026190766, 0.0719055271847999,
```

```
0.21640730222603838, 0.14106169740623525, 0.49811441302951087,
0.40449863737274727]
Updating best test loss: 3.14993


Starting epoch 11 / 50
Learning Rate for this epoch: 0.001
Epoch [11/50], Iter [50/209], Loss: total=2.708, reg=1.310,
containing_obj=0.816, no_obj=0.144, cls=0.438
Epoch [11/50], Iter [100/209], Loss: total=2.716, reg=1.325,
containing_obj=0.829, no_obj=0.141, cls=0.422
Epoch [11/50], Iter [150/209], Loss: total=2.742, reg=1.334,
containing_obj=0.831, no_obj=0.140, cls=0.438
Epoch [11/50], Iter [200/209], Loss: total=2.757, reg=1.354,
containing_obj=0.825, no_obj=0.141, cls=0.438


Starting epoch 12 / 50
Learning Rate for this epoch: 0.001
Epoch [12/50], Iter [50/209], Loss: total=2.700, reg=1.327,
containing_obj=0.821, no_obj=0.142, cls=0.410
Epoch [12/50], Iter [100/209], Loss: total=2.685, reg=1.328,
containing_obj=0.803, no_obj=0.141, cls=0.413
Epoch [12/50], Iter [150/209], Loss: total=2.671, reg=1.319,
containing_obj=0.805, no_obj=0.140, cls=0.406
Epoch [12/50], Iter [200/209], Loss: total=2.682, reg=1.311,
containing_obj=0.819, no_obj=0.139, cls=0.413


Starting epoch 13 / 50
Learning Rate for this epoch: 0.001
Epoch [13/50], Iter [50/209], Loss: total=2.524, reg=1.224,
containing_obj=0.795, no_obj=0.138, cls=0.368
Epoch [13/50], Iter [100/209], Loss: total=2.596, reg=1.255,
containing_obj=0.820, no_obj=0.135, cls=0.385
Epoch [13/50], Iter [150/209], Loss: total=2.643, reg=1.286,
containing_obj=0.839, no_obj=0.134, cls=0.384
Epoch [13/50], Iter [200/209], Loss: total=2.648, reg=1.295,
containing_obj=0.831, no_obj=0.134, cls=0.389
Updating best test loss: 3.03751


Starting epoch 14 / 50
Learning Rate for this epoch: 0.001
Epoch [14/50], Iter [50/209], Loss: total=2.533, reg=1.245,
containing_obj=0.777, no_obj=0.141, cls=0.370
Epoch [14/50], Iter [100/209], Loss: total=2.480, reg=1.214,
containing_obj=0.777, no_obj=0.142, cls=0.347
```

```
Epoch [14/50], Iter [150/209], Loss: total=2.520, reg=1.227,
containing_obj=0.797, no_obj=0.143, cls=0.354
Epoch [14/50], Iter [200/209], Loss: total=2.537, reg=1.231,
containing_obj=0.802, no_obj=0.142, cls=0.362


Starting epoch 15 / 50
Learning Rate for this epoch: 0.001
Epoch [15/50], Iter [50/209], Loss: total=2.447, reg=1.170,
containing_obj=0.820, no_obj=0.134, cls=0.323
Epoch [15/50], Iter [100/209], Loss: total=2.434, reg=1.165,
containing_obj=0.806, no_obj=0.140, cls=0.324
Epoch [15/50], Iter [150/209], Loss: total=2.458, reg=1.177,
containing_obj=0.813, no_obj=0.137, cls=0.330
Epoch [15/50], Iter [200/209], Loss: total=2.520, reg=1.217,
containing_obj=0.831, no_obj=0.136, cls=0.337
---Evaluate model on test samples---

100%|     | 4950/4950 [02:24<00:00, 34.36it/s]

---class aeroplane ap 0.4356732454051556---
---class bicycle ap 0.44810200781632786---
---class bird ap 0.3801434682582944---
---class boat ap 0.29092424903022696---
---class bottle ap 0.12055072478432052---
---class bus ap 0.5441086483550813---
---class car ap 0.5993128759021323---
---class cat ap 0.5496272609274507---
---class chair ap 0.20023555237683116---
---class cow ap 0.49366906046884235---
---class diningtable ap 0.14186838954549316---
---class dog ap 0.5555851985361089---
---class horse ap 0.5968196230312743---
---class motorbike ap 0.4222586849264899---
---class person ap 0.37512585571712664---
---class pottedplant ap 0.1738742044375341---
---class sheep ap 0.416266436233785---
---class sofa ap 0.30789329560014533---
---class train ap 0.5761414438038914---
---class tvmonitor ap 0.36003730538745715---
---map 0.3994108765271985---
14 [0.4356732454051556, 0.44810200781632786, 0.3801434682582944,
0.29092424903022696, 0.12055072478432052, 0.5441086483550813,
0.5993128759021323, 0.5496272609274507, 0.20023555237683116,
0.49366906046884235, 0.14186838954549316, 0.5555851985361089,
0.5968196230312743, 0.4222586849264899, 0.37512585571712664, 0.1738742044375341,
0.416266436233785, 0.30789329560014533, 0.5761414438038914, 0.36003730538745715]
```

```
Starting epoch 16 / 50
Learning Rate for this epoch: 0.001
Epoch [16/50], Iter [50/209], Loss: total=2.463, reg=1.200,
containing_obj=0.814, no_obj=0.141, cls=0.308
Epoch [16/50], Iter [100/209], Loss: total=2.486, reg=1.211,
containing_obj=0.834, no_obj=0.135, cls=0.306
Epoch [16/50], Iter [150/209], Loss: total=2.447, reg=1.183,
containing_obj=0.818, no_obj=0.137, cls=0.309
Epoch [16/50], Iter [200/209], Loss: total=2.456, reg=1.179,
containing_obj=0.826, no_obj=0.135, cls=0.315
Updating best test loss: 2.94462


Starting epoch 17 / 50
Learning Rate for this epoch: 0.001
Epoch [17/50], Iter [50/209], Loss: total=2.406, reg=1.157,
containing_obj=0.822, no_obj=0.140, cls=0.287
Epoch [17/50], Iter [100/209], Loss: total=2.414, reg=1.163,
containing_obj=0.819, no_obj=0.144, cls=0.288
Epoch [17/50], Iter [150/209], Loss: total=2.398, reg=1.142,
containing_obj=0.821, no_obj=0.142, cls=0.293
Epoch [17/50], Iter [200/209], Loss: total=2.408, reg=1.147,
containing_obj=0.823, no_obj=0.140, cls=0.298


Starting epoch 18 / 50
Learning Rate for this epoch: 0.001
Epoch [18/50], Iter [50/209], Loss: total=2.487, reg=1.203,
containing_obj=0.847, no_obj=0.137, cls=0.300
Epoch [18/50], Iter [100/209], Loss: total=2.434, reg=1.156,
containing_obj=0.836, no_obj=0.139, cls=0.302
Epoch [18/50], Iter [150/209], Loss: total=2.404, reg=1.140,
containing_obj=0.833, no_obj=0.137, cls=0.293
Epoch [18/50], Iter [200/209], Loss: total=2.401, reg=1.137,
containing_obj=0.833, no_obj=0.137, cls=0.293


Starting epoch 19 / 50
Learning Rate for this epoch: 0.001
Epoch [19/50], Iter [50/209], Loss: total=2.349, reg=1.104,
containing_obj=0.829, no_obj=0.132, cls=0.284
Epoch [19/50], Iter [100/209], Loss: total=2.315, reg=1.100,
containing_obj=0.808, no_obj=0.136, cls=0.271
Epoch [19/50], Iter [150/209], Loss: total=2.338, reg=1.111,
containing_obj=0.814, no_obj=0.140, cls=0.272
Epoch [19/50], Iter [200/209], Loss: total=2.334, reg=1.105,
containing_obj=0.815, no_obj=0.141, cls=0.272
Updating best test loss: 2.92473
```

```
Starting epoch 20 / 50
Learning Rate for this epoch: 0.001
Epoch [20/50], Iter [50/209], Loss: total=2.296, reg=1.087,
containing_obj=0.826, no_obj=0.132, cls=0.251
Epoch [20/50], Iter [100/209], Loss: total=2.329, reg=1.105,
containing_obj=0.832, no_obj=0.136, cls=0.256
Epoch [20/50], Iter [150/209], Loss: total=2.308, reg=1.094,
containing_obj=0.824, no_obj=0.137, cls=0.252
Epoch [20/50], Iter [200/209], Loss: total=2.291, reg=1.086,
containing_obj=0.812, no_obj=0.138, cls=0.254
---Evaluate model on test samples---

100%|     | 4950/4950 [02:23<00:00, 34.58it/s]

---class aeroplane ap 0.4500290385934387---
---class bicycle ap 0.6119677548336837---
---class bird ap 0.47045234354917054---
---class boat ap 0.33807746224643087---
---class bottle ap 0.17938049648987325---
---class bus ap 0.5857919548744188---
---class car ap 0.6546783146730449---
---class cat ap 0.6193989618047662---
---class chair ap 0.2984238286421016---
---class cow ap 0.5704814432674307---
---class diningtable ap 0.1726416483340955---
---class dog ap 0.6256145581359478---
---class horse ap 0.6486127212155299---
---class motorbike ap 0.4805464038752212---
---class person ap 0.49900208745793134---
---class pottedplant ap 0.2053964101368331---
---class sheep ap 0.4259484914784117---
---class sofa ap 0.4440582509298917---
---class train ap 0.6016929814373102---
---class tvmonitor ap 0.4807186224829285---
---map 0.46814568872292306---
19 [0.4500290385934387, 0.6119677548336837, 0.47045234354917054,
0.33807746224643087, 0.17938049648987325, 0.5857919548744188,
0.6546783146730449, 0.6193989618047662, 0.2984238286421016, 0.5704814432674307,
0.1726416483340955, 0.6256145581359478, 0.6486127212155299, 0.4805464038752212,
0.49900208745793134, 0.2053964101368331, 0.4259484914784117, 0.4440582509298917,
0.6016929814373102, 0.4807186224829285]
Updating best test loss: 2.91945


Starting epoch 21 / 50
Learning Rate for this epoch: 0.001
Epoch [21/50], Iter [50/209], Loss: total=2.316, reg=1.075,
```

```
containing_obj=0.848, no_obj=0.137, cls=0.256
Epoch [21/50], Iter [100/209], Loss: total=2.253, reg=1.031,
containing_obj=0.825, no_obj=0.139, cls=0.258
Epoch [21/50], Iter [150/209], Loss: total=2.287, reg=1.053,
containing_obj=0.832, no_obj=0.139, cls=0.264
Epoch [21/50], Iter [200/209], Loss: total=2.274, reg=1.062,
containing_obj=0.821, no_obj=0.137, cls=0.255


Starting epoch 22 / 50
Learning Rate for this epoch: 0.001
Epoch [22/50], Iter [50/209], Loss: total=2.252, reg=1.053,
containing_obj=0.814, no_obj=0.143, cls=0.242
Epoch [22/50], Iter [100/209], Loss: total=2.232, reg=1.035,
containing_obj=0.816, no_obj=0.143, cls=0.238
Epoch [22/50], Iter [150/209], Loss: total=2.239, reg=1.040,
containing_obj=0.821, no_obj=0.140, cls=0.238
Epoch [22/50], Iter [200/209], Loss: total=2.231, reg=1.039,
containing_obj=0.814, no_obj=0.142, cls=0.236
Updating best test loss: 2.87325


Starting epoch 23 / 50
Learning Rate for this epoch: 0.001
Epoch [23/50], Iter [50/209], Loss: total=2.135, reg=0.992,
containing_obj=0.770, no_obj=0.151, cls=0.221
Epoch [23/50], Iter [100/209], Loss: total=2.119, reg=1.002,
containing_obj=0.749, no_obj=0.154, cls=0.215
Epoch [23/50], Iter [150/209], Loss: total=2.148, reg=1.006,
containing_obj=0.750, no_obj=0.161, cls=0.230
Epoch [23/50], Iter [200/209], Loss: total=2.157, reg=1.021,
containing_obj=0.738, no_obj=0.167, cls=0.230
Updating best test loss: 2.85476


Starting epoch 24 / 50
Learning Rate for this epoch: 0.001
Epoch [24/50], Iter [50/209], Loss: total=2.188, reg=1.058,
containing_obj=0.676, no_obj=0.205, cls=0.248
Epoch [24/50], Iter [100/209], Loss: total=2.099, reg=1.018,
containing_obj=0.646, no_obj=0.207, cls=0.227
Epoch [24/50], Iter [150/209], Loss: total=2.079, reg=1.005,
containing_obj=0.639, no_obj=0.208, cls=0.226
Epoch [24/50], Iter [200/209], Loss: total=2.075, reg=1.004,
containing_obj=0.641, no_obj=0.207, cls=0.222
Updating best test loss: 2.76584
```

```
Starting epoch 25 / 50
Learning Rate for this epoch: 0.001
Epoch [25/50], Iter [50/209], Loss: total=2.002, reg=0.976,
containing_obj=0.618, no_obj=0.209, cls=0.199
Epoch [25/50], Iter [100/209], Loss: total=2.063, reg=1.007,
containing_obj=0.640, no_obj=0.209, cls=0.206
Epoch [25/50], Iter [150/209], Loss: total=2.033, reg=0.988,
containing_obj=0.627, no_obj=0.207, cls=0.211
Epoch [25/50], Iter [200/209], Loss: total=2.028, reg=0.985,
containing_obj=0.624, no_obj=0.209, cls=0.210
---Evaluate model on test samples---

100%|     | 4950/4950 [02:09<00:00, 38.27it/s]

---class aeroplane ap 0.4493943018533752---
---class bicycle ap 0.5815049295793329---
---class bird ap 0.40438008537311365---
---class boat ap 0.3059144934344722---
---class bottle ap 0.1880503664061213---
---class bus ap 0.5294113446256266---
---class car ap 0.6277563886035822---
---class cat ap 0.6563460194727262---
---class chair ap 0.2818475300113292---
---class cow ap 0.46979663971899654---
---class diningtable ap 0.2724369353427165---
---class dog ap 0.6138955865319873---
---class horse ap 0.6640598729179764---
---class motorbike ap 0.48992167955986476---
---class person ap 0.5380842520165102---
---class pottedplant ap 0.15371697792721345---
---class sheep ap 0.4520906661645119---
---class sofa ap 0.4427459560395489---
---class train ap 0.5235328183834421---
---class tvmonitor ap 0.46897552682954124---
---map 0.45569311853959943---
24 [0.4493943018533752, 0.5815049295793329, 0.40438008537311365,
0.3059144934344722, 0.1880503664061213, 0.5294113446256266, 0.6277563886035822,
0.6563460194727262, 0.2818475300113292, 0.46979663971899654, 0.2724369353427165,
0.6138955865319873, 0.6640598729179764, 0.48992167955986476, 0.5380842520165102,
0.15371697792721345, 0.4520906661645119, 0.4427459560395489, 0.5235328183834421,
0.46897552682954124]
Updating best test loss: 2.73554


Starting epoch 26 / 50
Learning Rate for this epoch: 0.001
Epoch [26/50], Iter [50/209], Loss: total=1.988, reg=0.970,
containing_obj=0.620, no_obj=0.206, cls=0.192
Epoch [26/50], Iter [100/209], Loss: total=2.002, reg=0.961,
```

```
containing_obj=0.625, no_obj=0.207, cls=0.209
Epoch [26/50], Iter [150/209], Loss: total=2.005, reg=0.967,
containing_obj=0.618, no_obj=0.209, cls=0.211
Epoch [26/50], Iter [200/209], Loss: total=1.999, reg=0.962,
containing_obj=0.612, no_obj=0.208, cls=0.217


Starting epoch 27 / 50
Learning Rate for this epoch: 0.001
Epoch [27/50], Iter [50/209], Loss: total=1.905, reg=0.920,
containing_obj=0.591, no_obj=0.201, cls=0.194
Epoch [27/50], Iter [100/209], Loss: total=1.956, reg=0.936,
containing_obj=0.614, no_obj=0.207, cls=0.200
Epoch [27/50], Iter [150/209], Loss: total=1.966, reg=0.946,
containing_obj=0.610, no_obj=0.207, cls=0.203
Epoch [27/50], Iter [200/209], Loss: total=1.973, reg=0.952,
containing_obj=0.607, no_obj=0.208, cls=0.206


Starting epoch 28 / 50
Learning Rate for this epoch: 0.001
Epoch [28/50], Iter [50/209], Loss: total=1.973, reg=0.961,
containing_obj=0.606, no_obj=0.203, cls=0.203
Epoch [28/50], Iter [100/209], Loss: total=1.901, reg=0.931,
containing_obj=0.580, no_obj=0.201, cls=0.189
Epoch [28/50], Iter [150/209], Loss: total=1.906, reg=0.939,
containing_obj=0.581, no_obj=0.201, cls=0.185
Epoch [28/50], Iter [200/209], Loss: total=1.932, reg=0.954,
containing_obj=0.591, no_obj=0.200, cls=0.187


Starting epoch 29 / 50
Learning Rate for this epoch: 0.001
Epoch [29/50], Iter [50/209], Loss: total=1.883, reg=0.899,
containing_obj=0.585, no_obj=0.205, cls=0.194
Epoch [29/50], Iter [100/209], Loss: total=1.848, reg=0.889,
containing_obj=0.576, no_obj=0.202, cls=0.180
Epoch [29/50], Iter [150/209], Loss: total=1.889, reg=0.917,
containing_obj=0.583, no_obj=0.203, cls=0.186
Epoch [29/50], Iter [200/209], Loss: total=1.884, reg=0.916,
containing_obj=0.580, no_obj=0.204, cls=0.185


Starting epoch 30 / 50
Learning Rate for this epoch: 0.001
Epoch [30/50], Iter [50/209], Loss: total=1.783, reg=0.856,
containing_obj=0.565, no_obj=0.201, cls=0.161
Epoch [30/50], Iter [100/209], Loss: total=1.828, reg=0.871,
```

```
containing_obj=0.573, no_obj=0.205, cls=0.180
Epoch [30/50], Iter [150/209], Loss: total=1.835, reg=0.875,
containing_obj=0.571, no_obj=0.209, cls=0.180
Epoch [30/50], Iter [200/209], Loss: total=1.849, reg=0.882,
containing_obj=0.575, no_obj=0.208, cls=0.184
---Evaluate model on test samples---

100%|     | 4950/4950 [02:11<00:00, 37.68it/s]

---class aeroplane ap 0.4370347377431284---
---class bicycle ap 0.6063564896750449---
---class bird ap 0.4513877136383035---
---class boat ap 0.3177819913946993---
---class bottle ap 0.22718827519784496---
---class bus ap 0.6232368912901032---
---class car ap 0.6722871253118491---
---class cat ap 0.5940839534050067---
---class chair ap 0.32175522703517934---
---class cow ap 0.5520750850568417---
---class diningtable ap 0.2596434021904313---
---class dog ap 0.5652328641056429---
---class horse ap 0.6683811709784176---
---class motorbike ap 0.5780252501379156---
---class person ap 0.5511284046251623---
---class pottedplant ap 0.20948417858732127---
---class sheep ap 0.44027965034567046---
---class sofa ap 0.4079782852830158---
---class train ap 0.5780049653813829---
---class tvmonitor ap 0.5218785526769412---
---map 0.47916121070299517---
29 [0.4370347377431284, 0.6063564896750449, 0.4513877136383035,
0.3177819913946993, 0.22718827519784496, 0.6232368912901032, 0.6722871253118491,
0.5940839534050067, 0.32175522703517934, 0.5520750850568417, 0.2596434021904313,
0.5652328641056429, 0.6683811709784176, 0.5780252501379156, 0.5511284046251623,
0.20948417858732127, 0.44027965034567046, 0.4079782852830158,
0.5780049653813829, 0.5218785526769412]


Starting epoch 31 / 50
Learning Rate for this epoch: 0.0001
Epoch [31/50], Iter [50/209], Loss: total=1.789, reg=0.856,
containing_obj=0.585, no_obj=0.200, cls=0.148
Epoch [31/50], Iter [100/209], Loss: total=1.808, reg=0.866,
containing_obj=0.591, no_obj=0.200, cls=0.152
Epoch [31/50], Iter [150/209], Loss: total=1.788, reg=0.854,
containing_obj=0.571, no_obj=0.203, cls=0.160
Epoch [31/50], Iter [200/209], Loss: total=1.784, reg=0.856,
containing_obj=0.564, no_obj=0.203, cls=0.161
Updating best test loss: 2.66404
```

```
Starting epoch 32 / 50
Learning Rate for this epoch: 0.0001
Epoch [32/50], Iter [50/209], Loss: total=1.718, reg=0.824,
containing_obj=0.540, no_obj=0.207, cls=0.147
Epoch [32/50], Iter [100/209], Loss: total=1.700, reg=0.817,
containing_obj=0.533, no_obj=0.206, cls=0.143
Epoch [32/50], Iter [150/209], Loss: total=1.739, reg=0.833,
containing_obj=0.549, no_obj=0.203, cls=0.153
Epoch [32/50], Iter [200/209], Loss: total=1.726, reg=0.825,
containing_obj=0.546, no_obj=0.204, cls=0.152


Starting epoch 33 / 50
Learning Rate for this epoch: 0.0001
Epoch [33/50], Iter [50/209], Loss: total=1.736, reg=0.837,
containing_obj=0.546, no_obj=0.194, cls=0.158
Epoch [33/50], Iter [100/209], Loss: total=1.690, reg=0.812,
containing_obj=0.529, no_obj=0.201, cls=0.147
Epoch [33/50], Iter [150/209], Loss: total=1.694, reg=0.809,
containing_obj=0.536, no_obj=0.203, cls=0.146
Epoch [33/50], Iter [200/209], Loss: total=1.689, reg=0.800,
containing_obj=0.541, no_obj=0.205, cls=0.143
Updating best test loss: 2.66111


Starting epoch 34 / 50
Learning Rate for this epoch: 0.0001
Epoch [34/50], Iter [50/209], Loss: total=1.700, reg=0.798,
containing_obj=0.551, no_obj=0.204, cls=0.146
Epoch [34/50], Iter [100/209], Loss: total=1.686, reg=0.789,
containing_obj=0.544, no_obj=0.207, cls=0.146
Epoch [34/50], Iter [150/209], Loss: total=1.667, reg=0.781,
containing_obj=0.539, no_obj=0.205, cls=0.142
Epoch [34/50], Iter [200/209], Loss: total=1.667, reg=0.785,
containing_obj=0.535, no_obj=0.208, cls=0.139
Updating best test loss: 2.65856


Starting epoch 35 / 50
Learning Rate for this epoch: 0.0001
Epoch [35/50], Iter [50/209], Loss: total=1.629, reg=0.779,
containing_obj=0.529, no_obj=0.202, cls=0.119
Epoch [35/50], Iter [100/209], Loss: total=1.646, reg=0.785,
containing_obj=0.530, no_obj=0.202, cls=0.128
Epoch [35/50], Iter [150/209], Loss: total=1.638, reg=0.776,
containing_obj=0.529, no_obj=0.204, cls=0.129
```

```
Epoch [35/50], Iter [200/209], Loss: total=1.652, reg=0.782,
containing_obj=0.532, no_obj=0.205, cls=0.133
---Evaluate model on test samples---

100%|     | 4950/4950 [02:26<00:00, 33.73it/s]

---class aeroplane ap 0.5124813531391292---
---class bicycle ap 0.6406666728820745---
---class bird ap 0.5267294913528213---
---class boat ap 0.3520827537300927---
---class bottle ap 0.23707125023635497---
---class bus ap 0.6381050036675867---
---class car ap 0.6898492314049264---
---class cat ap 0.6869430381256758---
---class chair ap 0.32649655488905494---
---class cow ap 0.5229444279305109---
---class diningtable ap 0.32013941508431654---
---class dog ap 0.6362180706662386---
---class horse ap 0.6834630980947245---
---class motorbike ap 0.6141312524877574---
---class person ap 0.5667527822081906---
---class pottedplant ap 0.22793007147545105---
---class sheep ap 0.46117512972559727---
---class sofa ap 0.4738739595996908---
---class train ap 0.6174561207624698---
---class tvmonitor ap 0.5261043183866694---
---map 0.5130306997924666---
34 [0.5124813531391292, 0.6406666728820745, 0.5267294913528213,
0.3520827537300927, 0.23707125023635497, 0.6381050036675867, 0.6898492314049264,
0.6869430381256758, 0.32649655488905494, 0.5229444279305109,
0.32013941508431654, 0.6362180706662386, 0.6834630980947245, 0.6141312524877574,
0.5667527822081906, 0.22793007147545105, 0.46117512972559727,
0.4738739595996908, 0.6174561207624698, 0.5261043183866694]


Starting epoch 36 / 50
Learning Rate for this epoch: 0.0001
Epoch [36/50], Iter [50/209], Loss: total=1.661, reg=0.783,
containing_obj=0.522, no_obj=0.214, cls=0.141
Epoch [36/50], Iter [100/209], Loss: total=1.698, reg=0.798,
containing_obj=0.542, no_obj=0.212, cls=0.145
Epoch [36/50], Iter [150/209], Loss: total=1.685, reg=0.790,
containing_obj=0.537, no_obj=0.210, cls=0.147
Epoch [36/50], Iter [200/209], Loss: total=1.663, reg=0.782,
containing_obj=0.532, no_obj=0.208, cls=0.140


Starting epoch 37 / 50
Learning Rate for this epoch: 0.0001
```

```
Epoch [37/50], Iter [50/209], Loss: total=1.622, reg=0.756,
containing_obj=0.531, no_obj=0.206, cls=0.130
Epoch [37/50], Iter [100/209], Loss: total=1.586, reg=0.736,
containing_obj=0.517, no_obj=0.208, cls=0.125
Epoch [37/50], Iter [150/209], Loss: total=1.590, reg=0.738,
containing_obj=0.517, no_obj=0.206, cls=0.128
Epoch [37/50], Iter [200/209], Loss: total=1.609, reg=0.755,
containing_obj=0.520, no_obj=0.206, cls=0.128
Updating best test loss: 2.65273


Starting epoch 38 / 50
Learning Rate for this epoch: 0.0001
Epoch [38/50], Iter [50/209], Loss: total=1.565, reg=0.728,
containing_obj=0.507, no_obj=0.204, cls=0.125
Epoch [38/50], Iter [100/209], Loss: total=1.584, reg=0.736,
containing_obj=0.519, no_obj=0.201, cls=0.129
Epoch [38/50], Iter [150/209], Loss: total=1.594, reg=0.744,
containing_obj=0.517, no_obj=0.205, cls=0.127
Epoch [38/50], Iter [200/209], Loss: total=1.604, reg=0.752,
containing_obj=0.517, no_obj=0.206, cls=0.129


Starting epoch 39 / 50
Learning Rate for this epoch: 0.0001
Epoch [39/50], Iter [50/209], Loss: total=1.599, reg=0.757,
containing_obj=0.521, no_obj=0.199, cls=0.122
Epoch [39/50], Iter [100/209], Loss: total=1.596, reg=0.756,
containing_obj=0.520, no_obj=0.200, cls=0.122
Epoch [39/50], Iter [150/209], Loss: total=1.592, reg=0.752,
containing_obj=0.515, no_obj=0.205, cls=0.121
Epoch [39/50], Iter [200/209], Loss: total=1.599, reg=0.752,
containing_obj=0.516, no_obj=0.209, cls=0.121


Starting epoch 40 / 50
Learning Rate for this epoch: 0.0001
Epoch [40/50], Iter [50/209], Loss: total=1.646, reg=0.776,
containing_obj=0.539, no_obj=0.200, cls=0.131
Epoch [40/50], Iter [100/209], Loss: total=1.586, reg=0.747,
containing_obj=0.514, no_obj=0.204, cls=0.122
Epoch [40/50], Iter [150/209], Loss: total=1.589, reg=0.748,
containing_obj=0.516, no_obj=0.204, cls=0.121
Epoch [40/50], Iter [200/209], Loss: total=1.598, reg=0.751,
containing_obj=0.516, no_obj=0.206, cls=0.125
---Evaluate model on test samples---

100%|     | 4950/4950 [02:19<00:00, 35.40it/s]
```

```
---class aeroplane ap 0.5041299236726144---
---class bicycle ap 0.6702644041146242---
---class bird ap 0.5443064755457228---
---class boat ap 0.34595750439014633---
---class bottle ap 0.24733893947292582---
---class bus ap 0.6477820421617064---
---class car ap 0.6934408739149409---
---class cat ap 0.6793073088527579---
---class chair ap 0.3329980944348795---
---class cow ap 0.5591092977325537---
---class diningtable ap 0.3303599087870781---
---class dog ap 0.6339171482342213---
---class horse ap 0.7086375921419621---
---class motorbike ap 0.6158403105188133---
---class person ap 0.565730483438492---
---class pottedplant ap 0.22363083087776076---
---class sheep ap 0.5013237735445348---
---class sofa ap 0.4762274064285803---
---class train ap 0.6200301629226648---
---class tvmonitor ap 0.5077442503051058---
---map 0.5204038365746043---
39 [0.5041299236726144, 0.6702644041146242, 0.5443064755457228,
0.34595750439014633, 0.24733893947292582, 0.6477820421617064,
0.6934408739149409, 0.6793073088527579, 0.3329980944348795, 0.5591092977325537,
0.3303599087870781, 0.6339171482342213, 0.7086375921419621, 0.6158403105188133,
0.565730483438492, 0.22363083087776076, 0.5013237735445348, 0.4762274064285803,
0.6200301629226648, 0.5077442503051058]
Updating best test loss: 2.63933


Starting epoch 41 / 50
Learning Rate for this epoch: 1e-05
Epoch [41/50], Iter [50/209], Loss: total=1.688, reg=0.812,
containing_obj=0.558, no_obj=0.200, cls=0.119
Epoch [41/50], Iter [100/209], Loss: total=1.604, reg=0.769,
containing_obj=0.523, no_obj=0.204, cls=0.109
Epoch [41/50], Iter [150/209], Loss: total=1.588, reg=0.756,
containing_obj=0.519, no_obj=0.204, cls=0.108
Epoch [41/50], Iter [200/209], Loss: total=1.588, reg=0.753,
containing_obj=0.517, no_obj=0.204, cls=0.114


Starting epoch 42 / 50
Learning Rate for this epoch: 1e-05
Epoch [42/50], Iter [50/209], Loss: total=1.491, reg=0.693,
containing_obj=0.490, no_obj=0.201, cls=0.108
Epoch [42/50], Iter [100/209], Loss: total=1.529, reg=0.716,
containing_obj=0.501, no_obj=0.196, cls=0.115
```

```
Epoch [42/50], Iter [150/209], Loss: total=1.536, reg=0.722,
containing_obj=0.502, no_obj=0.196, cls=0.117
Epoch [42/50], Iter [200/209], Loss: total=1.545, reg=0.722,
containing_obj=0.506, no_obj=0.199, cls=0.118


Starting epoch 43 / 50
Learning Rate for this epoch: 1e-05
Epoch [43/50], Iter [50/209], Loss: total=1.499, reg=0.708,
containing_obj=0.483, no_obj=0.207, cls=0.101
Epoch [43/50], Iter [100/209], Loss: total=1.561, reg=0.740,
containing_obj=0.503, no_obj=0.206, cls=0.112
Epoch [43/50], Iter [150/209], Loss: total=1.577, reg=0.750,
containing_obj=0.506, no_obj=0.201, cls=0.119
Epoch [43/50], Iter [200/209], Loss: total=1.573, reg=0.745,
containing_obj=0.505, no_obj=0.203, cls=0.120


Starting epoch 44 / 50
Learning Rate for this epoch: 1e-05
Epoch [44/50], Iter [50/209], Loss: total=1.470, reg=0.689,
containing_obj=0.467, no_obj=0.210, cls=0.106
Epoch [44/50], Iter [100/209], Loss: total=1.548, reg=0.732,
containing_obj=0.493, no_obj=0.205, cls=0.118
Epoch [44/50], Iter [150/209], Loss: total=1.556, reg=0.728,
containing_obj=0.504, no_obj=0.206, cls=0.118
Epoch [44/50], Iter [200/209], Loss: total=1.572, reg=0.737,
containing_obj=0.510, no_obj=0.205, cls=0.121


Starting epoch 45 / 50
Learning Rate for this epoch: 1e-05
Epoch [45/50], Iter [50/209], Loss: total=1.631, reg=0.771,
containing_obj=0.538, no_obj=0.202, cls=0.120
Epoch [45/50], Iter [100/209], Loss: total=1.594, reg=0.747,
containing_obj=0.527, no_obj=0.202, cls=0.118
Epoch [45/50], Iter [150/209], Loss: total=1.563, reg=0.728,
containing_obj=0.514, no_obj=0.206, cls=0.116
Epoch [45/50], Iter [200/209], Loss: total=1.556, reg=0.722,
containing_obj=0.514, no_obj=0.204, cls=0.116
---Evaluate model on test samples---
```

100%|      | 4950/4950 [02:18<00:00, 35.72it/s]

```
---class aeroplane ap 0.5281957531118149---
---class bicycle ap 0.6787577370933039---
---class bird ap 0.5471821781273337---
---class boat ap 0.371997979324474---
---class bottle ap 0.24145822286658025---
```

```
---class bus ap 0.6387569872644105---
---class car ap 0.6910533909333798---
---class cat ap 0.6815432789489099---
---class chair ap 0.3370886467266352---
---class cow ap 0.556894605082036---
---class diningtable ap 0.336280279594243---
---class dog ap 0.6324824447097587---
---class horse ap 0.691316028434873---
---class motorbike ap 0.6006904397326223---
---class person ap 0.57728022472045---
---class pottedplant ap 0.22064712850499352---
---class sheep ap 0.49075543672158295---
---class sofa ap 0.4660795964177883---
---class train ap 0.6189256732133483---
---class tvmonitor ap 0.5111932286065323---
---map 0.5209289630067535---
44 [0.5281957531118149, 0.6787577370933039, 0.5471821781273337,
0.371997979324474, 0.24145822286658025, 0.6387569872644105, 0.6910533909333798,
0.6815432789489099, 0.3370886467266352, 0.556894605082036, 0.336280279594243,
0.6324824447097587, 0.691316028434873, 0.6006904397326223, 0.57728022472045,
0.22064712850499352, 0.49075543672158295, 0.4660795964177883,
0.6189256732133483, 0.5111932286065323]


Starting epoch 46 / 50
Learning Rate for this epoch: 1e-05
Epoch [46/50], Iter [50/209], Loss: total=1.555, reg=0.733,
containing_obj=0.505, no_obj=0.197, cls=0.119
Epoch [46/50], Iter [100/209], Loss: total=1.548, reg=0.723,
containing_obj=0.503, no_obj=0.204, cls=0.118
Epoch [46/50], Iter [150/209], Loss: total=1.555, reg=0.726,
containing_obj=0.507, no_obj=0.204, cls=0.118
Epoch [46/50], Iter [200/209], Loss: total=1.562, reg=0.729,
containing_obj=0.513, no_obj=0.204, cls=0.116


Starting epoch 47 / 50
Learning Rate for this epoch: 1e-05
Epoch [47/50], Iter [50/209], Loss: total=1.532, reg=0.733,
containing_obj=0.483, no_obj=0.203, cls=0.112
Epoch [47/50], Iter [100/209], Loss: total=1.550, reg=0.728,
containing_obj=0.505, no_obj=0.200, cls=0.117
Epoch [47/50], Iter [150/209], Loss: total=1.554, reg=0.728,
containing_obj=0.507, no_obj=0.202, cls=0.117
Epoch [47/50], Iter [200/209], Loss: total=1.548, reg=0.721,
containing_obj=0.506, no_obj=0.203, cls=0.118
```

```
Starting epoch 48 / 50
Learning Rate for this epoch: 1e-05
Epoch [48/50], Iter [50/209], Loss: total=1.509, reg=0.699,
containing_obj=0.491, no_obj=0.209, cls=0.110
Epoch [48/50], Iter [100/209], Loss: total=1.557, reg=0.728,
containing_obj=0.504, no_obj=0.206, cls=0.119
Epoch [48/50], Iter [150/209], Loss: total=1.565, reg=0.731,
containing_obj=0.508, no_obj=0.206, cls=0.119
Epoch [48/50], Iter [200/209], Loss: total=1.566, reg=0.728,
containing_obj=0.510, no_obj=0.208, cls=0.120


Starting epoch 49 / 50
Learning Rate for this epoch: 1e-05
Epoch [49/50], Iter [50/209], Loss: total=1.544, reg=0.711,
containing_obj=0.511, no_obj=0.204, cls=0.118
Epoch [49/50], Iter [100/209], Loss: total=1.550, reg=0.722,
containing_obj=0.506, no_obj=0.200, cls=0.122
Epoch [49/50], Iter [150/209], Loss: total=1.558, reg=0.734,
containing_obj=0.504, no_obj=0.200, cls=0.120
Epoch [49/50], Iter [200/209], Loss: total=1.575, reg=0.745,
containing_obj=0.511, no_obj=0.200, cls=0.119


Starting epoch 50 / 50
Learning Rate for this epoch: 1e-05
Epoch [50/50], Iter [50/209], Loss: total=1.563, reg=0.728,
containing_obj=0.521, no_obj=0.202, cls=0.112
Epoch [50/50], Iter [100/209], Loss: total=1.527, reg=0.714,
containing_obj=0.498, no_obj=0.207, cls=0.108
Epoch [50/50], Iter [150/209], Loss: total=1.543, reg=0.718,
containing_obj=0.505, no_obj=0.205, cls=0.115
Epoch [50/50], Iter [200/209], Loss: total=1.538, reg=0.715,
containing_obj=0.502, no_obj=0.206, cls=0.115
---Evaluate model on test samples---
```

```
100%|     | 4950/4950 [02:19<00:00, 35.59it/s]
```

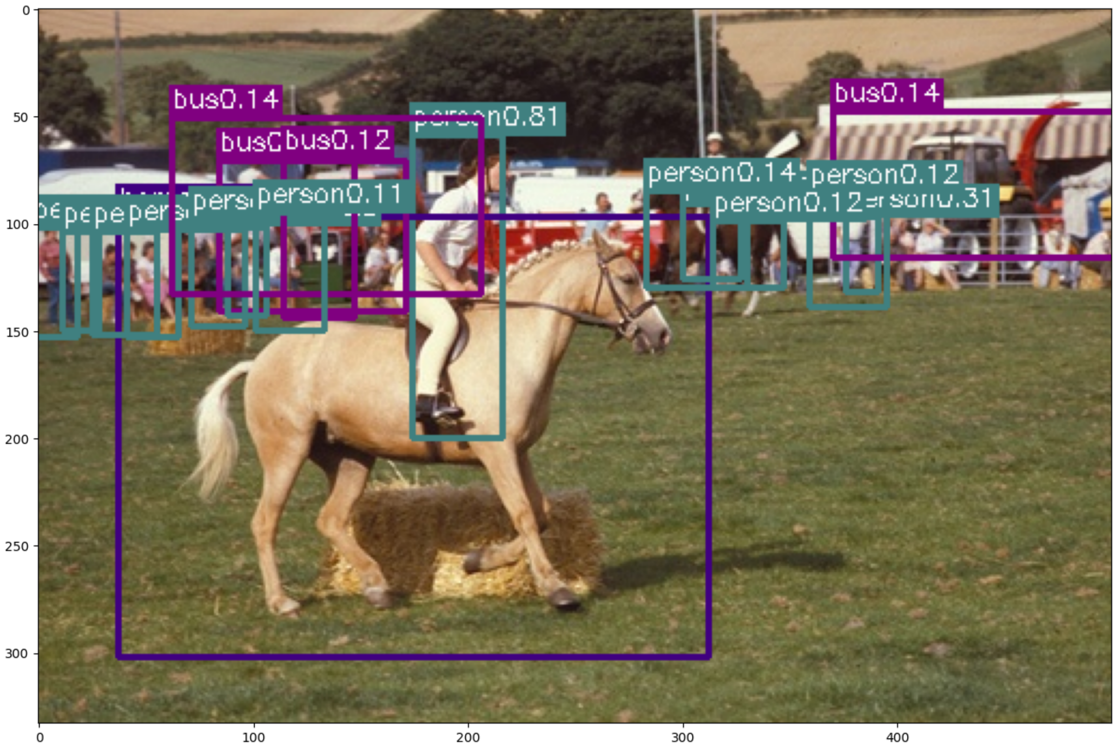
```
---class aeroplane ap 0.5313191239370854---
---class bicycle ap 0.6576073499532791---
---class bird ap 0.5366736391270497---
---class boat ap 0.37716124846787535---
---class bottle ap 0.24085737575341248---
---class bus ap 0.6485617792947089---
---class car ap 0.687869477840831---
---class cat ap 0.6731492633349193---
---class chair ap 0.3445235890792374---
---class cow ap 0.5548337618464732---
---class diningtable ap 0.350204457200313---
```

```
---class dog ap 0.6516651484153018---
---class horse ap 0.6954082953991244---
---class motorbike ap 0.6141076107499881---
---class person ap 0.5783747980777809---
---class pottedplant ap 0.22892791858007538---
---class sheep ap 0.4697638239596463---
---class sofa ap 0.4842278084485112---
---class train ap 0.6416915438594533---
---class tvmonitor ap 0.5286599705774611---
---map 0.5247793991951263---
49 [0.5313191239370854, 0.6576073499532791, 0.5366736391270497,
0.37716124846787535, 0.24085737575341248, 0.6485617792947089, 0.687869477840831,
0.6731492633349193, 0.3445235890792374, 0.5548337618464732, 0.350204457200313,
0.6516651484153018, 0.6954082953991244, 0.6141076107499881, 0.5783747980777809,
0.22892791858007538, 0.4697638239596463, 0.4842278084485112, 0.6416915438594533,
0.5286599705774611]
```

[ ]:
```python
### Print Prediction ###
net.eval()

# select random image from test set
image_name = random.choice(test_dataset.fnames)
image = cv2.imread(os.path.join(file_root_test, image_name))
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

print('predicting...')
result = predict_image(net, image_name, root_img_directory=file_root_test)
for left_up, right_bottom, class_name, _, prob in result:
    color = COLORS[VOC_CLASSES.index(class_name)]
    cv2.rectangle(image, left_up, right_bottom, color, 2)
    label = class_name + str(round(prob, 2))
    text_size, baseline = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.4, 1)
    p1 = (left_up[0], left_up[1] - text_size[1])
    cv2.rectangle(image, (p1[0] - 2 // 2, p1[1] - 2 - baseline), (p1[0] + text_size[0], p1[1] + text_size[1]),
                  color, -1)
    cv2.putText(image, label, (p1[0], p1[1] + baseline), cv2.FONT_HERSHEY_SIMPLEX, 0.4, (255, 255, 255), 1, 8)

plt.figure(figsize = (15,15))
plt.imshow(image)
```

predicting…

[ ]: <matplotlib.image.AxesImage at 0x151c66ea7a90>




[ ]:
```
### Evaluate on Test ###
test_aps = evaluate(net, test_dataset_file=annotation_file_test,
    img_root=file_root_test)
```

```
---Evaluate model on test samples---

100%|      | 4950/4950 [02:24<00:00, 34.37it/s]

---class aeroplane ap 0.5313191239370854---
---class bicycle ap 0.6576073499532791---
---class bird ap 0.5366736391270497---
---class boat ap 0.37716124846787535---
---class bottle ap 0.24085737575341248---
---class bus ap 0.6485617792947089---
---class car ap 0.687869477840831---
---class cat ap 0.6731492633349193---
---class chair ap 0.3445235890792374---
---class cow ap 0.5548337618464732---
---class diningtable ap 0.350204457200313---
---class dog ap 0.6516651484153018---
---class horse ap 0.6954082953991244---
---class motorbike ap 0.6141076107499881---
---class person ap 0.5783747980777809---
---class pottedplant ap 0.22892791858007538---
```

```
---class sheep ap 0.46976382395964763---
---class sofa ap 0.4842278084485112---
---class train ap 0.6416915438594533---
---class tvmonitor ap 0.5286599705774611---
---map 0.5247793991951263---
```

## 4 Save as PDF

[ ]:
```
%%capture

from google.colab import drive
drive.mount('/content/drive')
# install tex; first run may take several minutes
! apt-get install texlive-xetex
# file path and save location below are default; please change if they do not
  ↪match yours
! jupyter nbconvert --output-dir='/content/drive/MyDrive/' '/content/drive/
  ↪MyDrive/CS444/assignment3_starter_sp24/assignment3_part2/MP3_P2.ipynb' --to
  ↪pdf
```