Homework #8
Due M 4/22

You must turn in your code as well as output files. Please generate a report that contains the code and output in a single readable format.

Visit the book website to download companion software, including all the example problems.
`http://www.wiley.com/WileyCDA/WileyTitle/productCd-1118414322.html`

Test Images
Download the sample images from the class website
`http://www.ee.unlv.edu/~b1morris/ee482/docs/hw07`

1. Getting Started

   (a) (Recommended) Create a Github account if you do not have one. This is becoming more and more expected for (software) job interviews.

   (b) Install the `conda` environment management system. Choose either Anaconda (more space but pre-installs lots of useful ML packages) or Miniconda (smaller install) [Installation Instructions].

   (c) Create a environment. Default to basic the most up-to-date python by the command

   ```
   conda create --name myenv python
   ```

   On Windows, be sure you open the Anaconda prompt from the Start Menu. You can select either Anaconda Prompt or the PowerShell version.

   (d) Conda Cheatsheet

2. Deep Object Detection with Matlab

   The following code uses Matlab pretrained models for object detection. Some of the code may not work depending on your version of Matlab. For example YOLOv3 is not available on Matlab Online.

   Please note that all the code should be included in a single m file (delete YOLOv3 if `help yolov3ObjectDetector` returns not found).

   Notice that this code snipped uses %% to indicate Code Sections. You can quickly run an individual section of code using the Run Section (or Run and Advance) options in the Editor tooltip tab.

   You can play with different parameters by passing optional arguments to the detect function

   ```
   >> [___] = detect(___,Name,Value)
   ```

   You can find a list of different parameter names and values from the online help pages.

   - Fast-RCNN detect
   - SSD detect
   - YOLOv3 detect

   (a) Compare the detector performance on the `visionteam1.jpg` and `highway.png`. Put images side-by-side in a table with columns for the different detectors and rows for the image followed by the time to process the image. The table should be 3 columns and 4 rows. Comment on the differences in performance.

(b) Compare the results on the `dash02.mp4` file. What is the average processing time for each of the detectors. Comment on the differences in performance.

```
%% YOLOv3

%Specify the name of a pretrained YOLO v3 deep learning network.
name = 'tiny-yolov3-coco'; %or bigger model 'darknet53-coco'

%Create YOLO v3 object detector by using the pretrained YOLO v3
   network.
YOLO = yolov3ObjectDetector(name);

%Display and inspect the properties of the YOLO v3 object detector
   .
disp(YOLO)

%use analyzeNetwork to display the YOLO v3 network architecture
   and get information about the network layers. The network has
   two detection heads attached to the feature extraction network.
analyzeNetwork(YOLO.Network)

I = imread('visionteam1.jpg');
%put image into YOLO format
Iyolo = preprocess(YOLO, I);
%do detection
tic
[bboxes,scores,labels] = detect(YOLO,Iyolo,'DetectionPreprocessing
   ','none');
t_yolo = toc

detectedImg = insertObjectAnnotation(Iyolo,'Rectangle',bboxes,
   labels);
figure
imshow(detectedImg)


%% fast RCNN
data = load('fasterRCNNVehicleTrainingData.mat', 'detector');
detector_fastrcnn = data.detector;

%Display and inspect the properties of object detector.
disp(detector_fastrcnn)

%use analyzeNetwork to display the network architecture and get
   information about the network layers. The network has two
   detection heads attached to the feature extraction network.
analyzeNetwork(detector_fastrcnn.Network)

I = imread('highway.png');
```

```
39
40 tic
41 [bboxes,scores,labels] = detect(detector_fastrcnn,I);
42 t_fastrcnn = toc
43
44 if ~isempty(bboxes)
45     detectedI = insertObjectAnnotation(I,'rectangle',bboxes,
       cellstr(labels));
46 else
47    detectedI = insertText(I,[10 10],'No Detections');
48 end
49
50 figure
51 imshow(detectedI)
52
53 %% SSD
54 vehicleDetector = load('ssdVehicleDetector.mat','detector');
55 detector_ssd = vehicleDetector.detector;
56 I = imread('highway.png');
57
58 tic
59 [bboxes,scores,labels] = detect(detector_ssd,I);
60 t_ssd = toc
61
62 if ~isempty(bboxes)
63     detectedI = insertObjectAnnotation(I,'rectangle',bboxes,
       cellstr(labels));
64 else
65    detectedI = insertText(I,[10 10],'No Detections');
66 end
67
68 figure
69 imshow(detectedI)
70
71 %% dash video
72
73 %load from url
74 url = 'http://www.ee.unlv.edu/~b1morris/labs/cv_intro/dash02.mp4';
75 outfilename = websave('dash02.mp4', url);
76
77 videoSource = VideoReader('dash02.mp4');
78
79 vidout = uint8(zeros(240, 360, 3, 9));
80 t = zeros(1,9);          %processing time vector
81
82 %try out different set of frames to see performance.
83 index = [1, 9];
84 frames = read(videoSource, index);
85
```

```
for i = 1:9
    frame = frames(:,:,:,i);

    %crop image to center
    frame_crop = frame(240-120:240+120-1, 427-180:427+180-1,:);

        %[TODO be sure to try both the SSD and Fast RCNN models]
    tic
    [bboxes,scores,labels] = detect(detector_ssd,frame_crop);
    t(i) = toc;

    if ~isempty(bboxes)
        detectedI = insertObjectAnnotation(frame_crop,'rectangle',
           bboxes,cellstr(labels));
    else
        detectedI = insertText(frame_crop,[10 10],'No Detections')
           ;
    end

    vidout(:,:,:,i) = detectedI;
end

tmean = mean(t);
fps = 1/tmean;
fprintf(1, 'Average FPS = %2.2f (%2.4f s)\n', fps, tmean);


%% display vid output as frames
for i = 1:9
    figure
    imshow(vidout(:,:,:,i),[]);
    title(sprintf('frame = %d', index(1)+i-1));
end
```

3. Deep Object Detection with PyTorch

   In this problem, you will learn how to setup a virtual environment using Anaconda/miniconda for deep learning using Python. You will implement the YOLOv5 using the public repository from ultralytics. Look through the Quick Start Examples and closely follow Install, Inference, and Inference with detect.py sections.

   (a) Perform installation on your machine. It is recommended you use a Github account and clone as opposed to Downloading the Zip. Before any `pip` or `conda` package installation, be sure to create an environment (`yolov5`). You will only need to include python in your conda create command [link]. If you want to use a GPU you will need to modify the create command. Installation options for your OS can be found on the main PyTorch page.

   Please note that this code works best without any spaces in the path. On Windows you are encouraged to work in `C:\code` or similar folder.

   (b) Follow the Inference sample code to compare the results the `visionteam1.jpg` and `highway.png`.

Comment on the performance between images and by comparison with Matlab detectors. Add to your code and use `process_time` to get the time to process each image.

(c) Use `detect.py` to process the `dash02.mp4` video file. Show the same frames you used in the Matlab above and give the average processing time. Note you can look into `detect.py` to see the many command line options in the `parse_opt()` function. See the `run()` function for default values. You may want to trim the video to a smaller number of frames.

(d) Complete the tutorial Train Custom Data. Setup the necessary files to train the Tiny LISA traffic sign dataset (e.g. `db_tiny_lisa.ymal` etc.). You are not required to do the training (but you may). You must include your ymal file and the annotation file for `sample_001.jpg`.