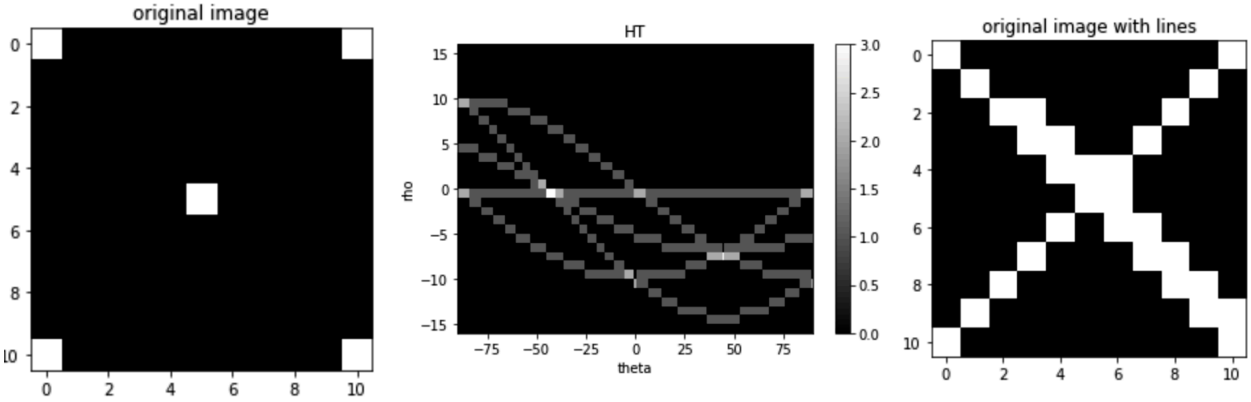
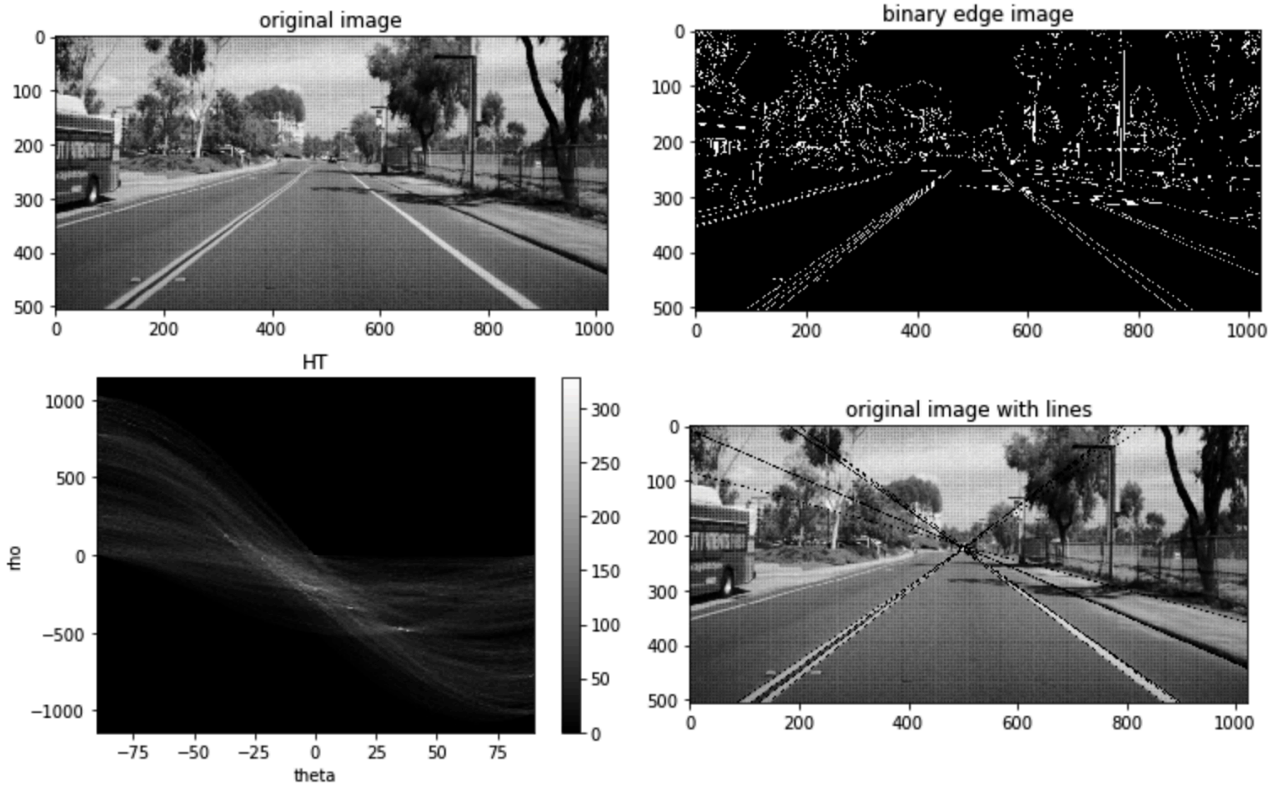
ECE253 HW4

problem 1

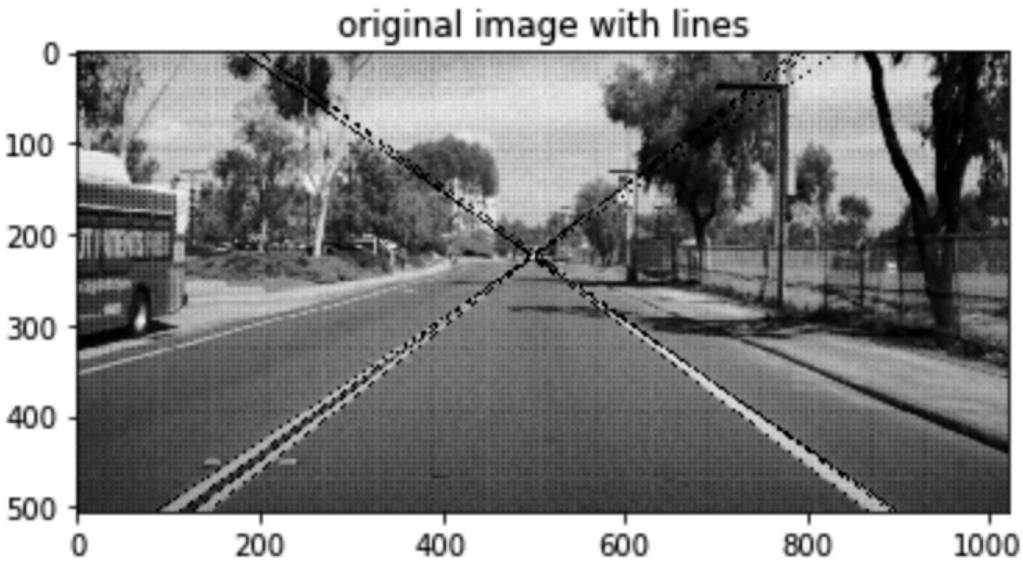
(ii) Below shows the original image, HT and original image with lines of question 2



(iii) Below shows the original image, binary edge imageHT and original image with lines of question 3

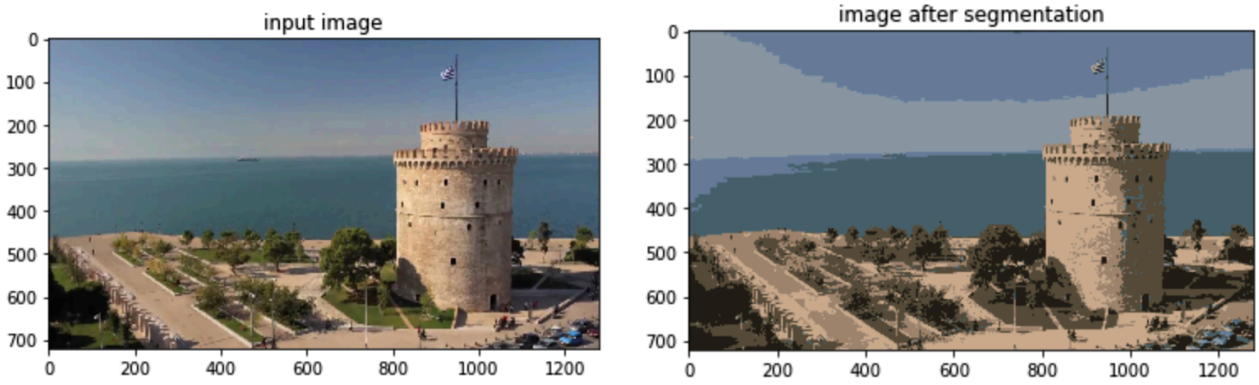


(iv) Below shows the original image with lines after thresholding θ. Here we take the range of θ in [-38, -34) and [34, 39).(The unit indicates here is degree)



problem 2

Input image and image after segmentation is shown as below



Centers of this picture are listed as below:

```
centers = [[ 73.05994068  99.2435464  109.60923826]
           [204.42738157 172.77634948 142.95925804]
           [159.97403578 134.15325946 110.56233517]
           [101.13762472 126.19515859 154.48806208]
           [138.70205064 152.95374115 165.88127351]
           [ 33.39527682  30.11767109  22.24859978]
           [ 86.71539553  77.17711212  57.43412537]]
```

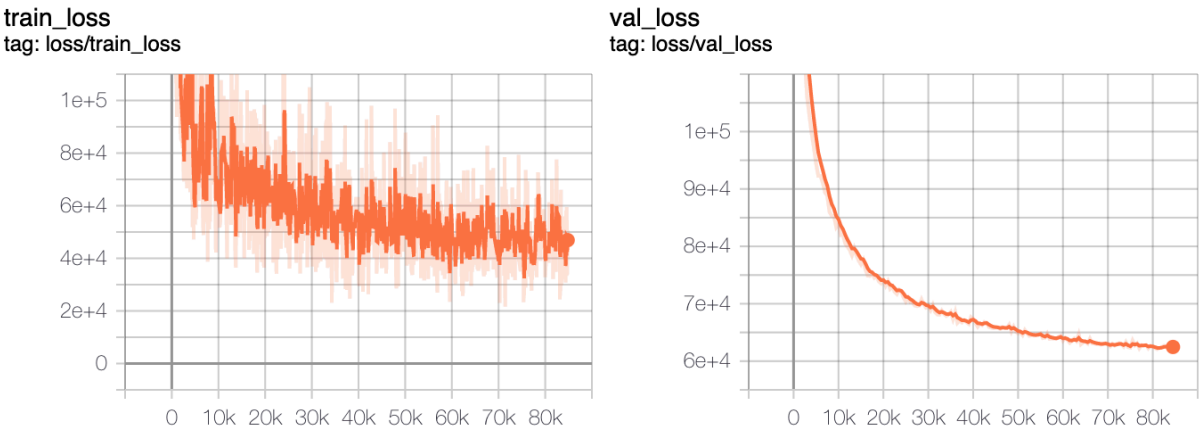
problem 3

(i) The model structure is as following: There are five convolution blocks in this FCN network and one classifier block. The first 2 blocks have 2 convolutional layers and 2 relu layers intersected with each other and one maxpooling layer, and the first block has covolutional layers of $3 \times 64 \times 3$ and $64 \times 64 \times 3$ and the second block has covolutional layers of $64 \times 128 \times 3$ and $128 \times 128 \times 3$. The other 3 blocks all have 3 convolutional layers and 3 relu layers intersected with each other and one maxpooling layer. The sizes of convolutional layers for the third block are one $128 \times 256 \times 3$ and two $256 \times 256 \times 3$. And the sizes of convolutional layes for the last two convolution blocks are two $512 \times 512 \times 3$, one with $256 \times 512 \times 3$ before it and one with $512 \times 512 \times 3$ before it. Then we use a classifier block with 3 convolutional layer with size $512 \times 4096 \times 7$, $4096 \times 4096 \times 1$, and $4096 \times 19 \times 3$, where 19 is the number of classes, and two relu layers and two dropout layers. Then we judge if this is a bilinear problem and use different score pooling methods.

(ii) We train the model from pre-trained model. As we can know from the following code that we pretrained the model with vgg16.

```
elif name in ["fcn32s", "fcn16s", "fcn8s"]:
    model = model(n_classes=n_classes, **param_dict)
    vgg16 = models.vgg16(pretrained=True)
    model.init_vgg16_params(vgg16)
```

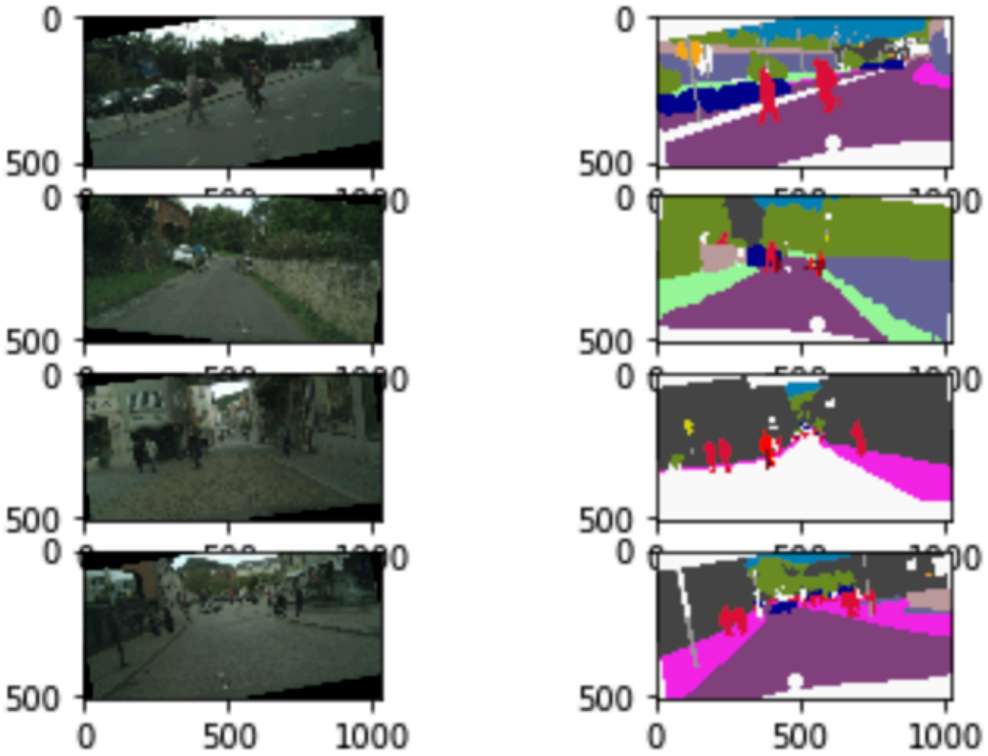
(iii) Here is the screenshot of curve for train and validation loss



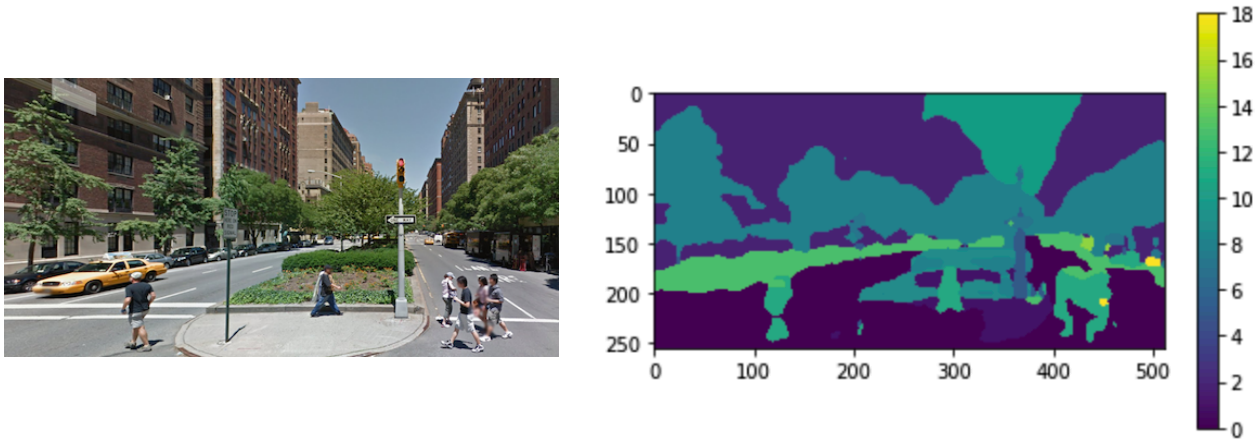
(iv) The metrics in the paper are four as below: Freqence Weight Accuracy, Mean Accuracy, Overall Accuracy and Mean IoU.

The one better is class 0 and the one worse is class 12

(v) Yes. Here I show one group of the labels and the predictions of images



(vi) Here is my test image and my result image run by myself. The result is noy good enough but can tell some rough ideas.



(vii) Maybe we can change the structure of our model to get better results like the number of layers and the size of each layers.