Homework #5
Due Th. 10/22

Be sure to show all your work for credit. You must turn in your code as well as output files (**code attached at the end of the report**).

Please generate a report that contains the code and output in a single readable format using Latex.

1. (GW 10.49)
2. Background Subtraction
   (a) Write a program to perform simple frame differencing on the video `traffic.avi` included in Matlab. Threshold the results to highlight the moving object pixels in a foreground image. Comment on the quality of object detection. Include the object image for frames 1, 41, and 91. If you do not have `traffic.avi` look to see if you have `viptraffic.avi`.
   (b) Repeat (a) but using a background image. The background image can be selected as the last frame of the video since no cars are present. Comment on the quality of the object detection. When does it work well, when does it fail?
   (c) Repeat with a background image obtained by taking the average of all the frames of the video. Display the background image in addition to the three output frames
   (d) Repeat with an adaptive background model. What value of α did you use to get good results. Display the foreground object images and the background image at frames 1, 41, 91. Again, comment on the quality of the foreground extraction.
3. Gaussian Mixture Model
   (a) Use Matlab's Computer Vision Toolbox to do foreground extraction using the Guassian mixture model. See `vision.ForegroundDetector`. Show the foreground for the frames 1, 41, 91.
   (b) The foreground image is noisy. Develop a clean-up routine to remove noise and smooth object boundaries. Show the foreground for the frames 1, 41, 91 and list the clean-up steps you performed.
   (c) Draw green bounding boxes around the vehicles in the orginal image and give the count of the number of vehicles in the top left corner with black text on a yellow background. Show the original frames 1, 41, 91 with overlayed bounding boxes.
4. Gaussian Mixture Model
   (a) Use Matlab's Computer Vision Toolbox to do optical flow calculation using Lucas and Kanade's algorithm. See `vision.OpticalFlow`. Show the original frames with flow arrows for the frames 1, 41, 91.
   (b) Repeat for the Horn Schunk algorithm and compare the results with LK.