Project 4 – Traffic Sign Recognition – Jiawei Ge

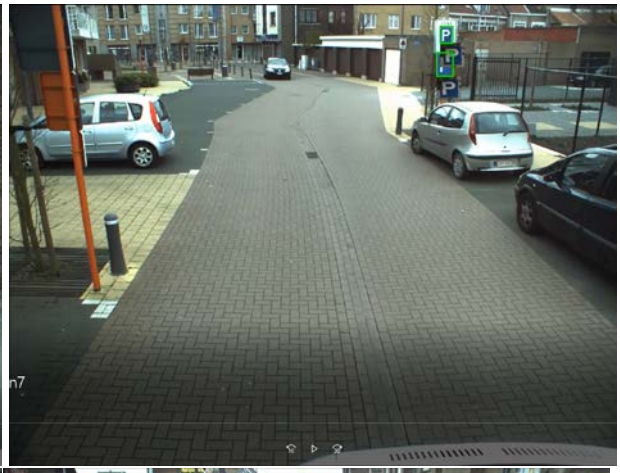
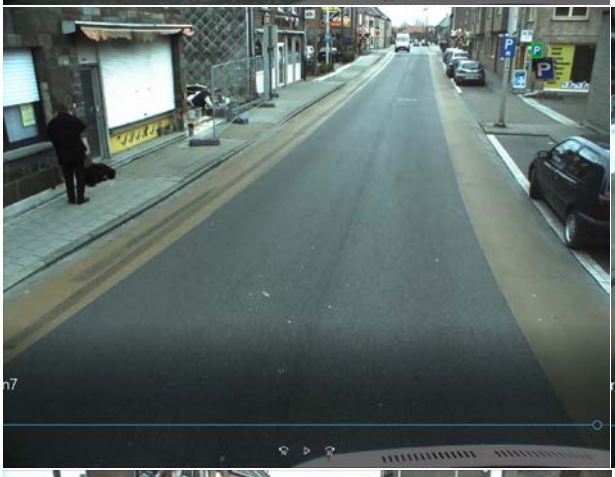
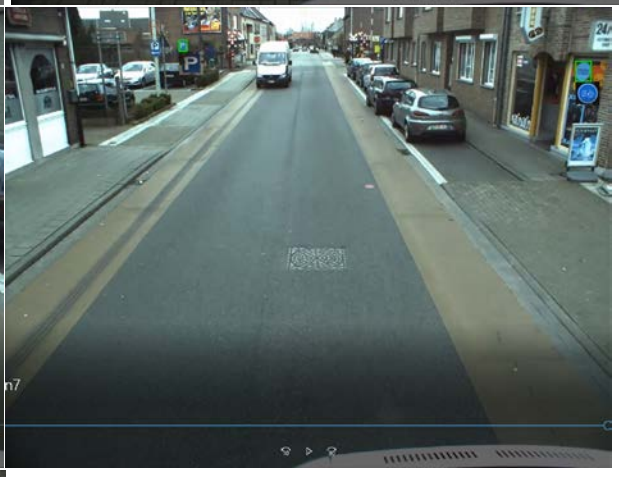
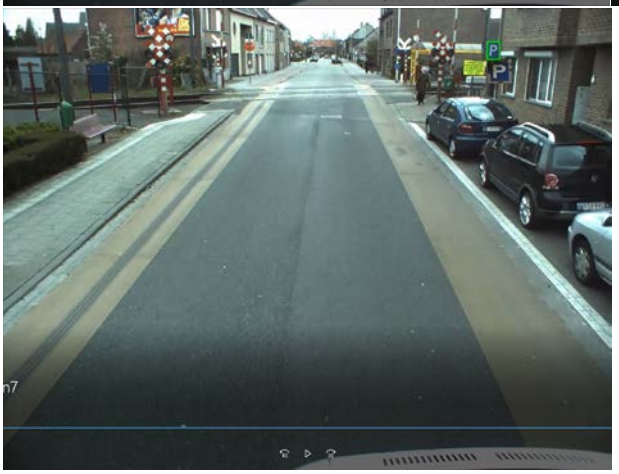
0. Output is a 95 seconds' video including all the 2861 frames.
1. Traffic Sign Detection. (HSV Approach)

   It's difficult to choose thresholds for HSV approach than MSER approach because the brightness of captured frames. However, the MSER approach will easily regard any flat areas like walls, posters, cars' headlights, etc. as traffic signs. The best approach should be the combination of these 2 methods. Time is the main issue I didn't do it.

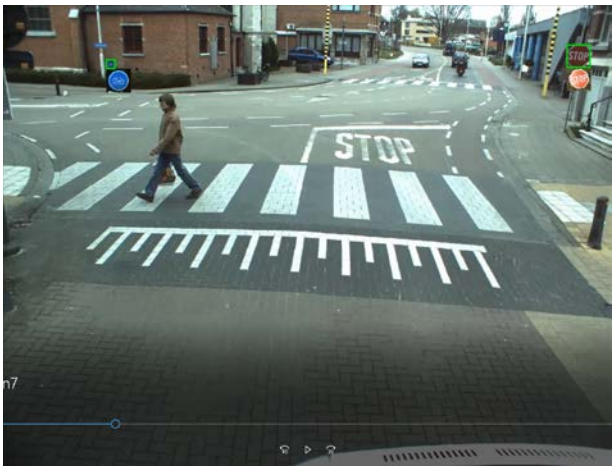
   1) Image denoise. *Imgaussfilt* used.
   2) Image normalization. Separate R, G, B channels and apply *stretchlim* & *imadjust* function to normalize image, then combine them again.
   3) HSV threshold setting. Based on the fact that the increase of the V value leads to the decrease of the S value which will cause an increase of H range for a specific color.
      i. 2 threshold combinations are enough for red color because red is easier to tell compared with blue: (H < 0.028 | H > 0.98) & S > 0.68 & V > 0.6 and (H < 0.02 | H > 0.95) & S > 0.6 & V > 0.15.
      ii. 4 threshold combinations are used for blue color because it's affected by brightness more: (H > 0.45 & H < 0.72 & S > 0.3 & V > 0.97) | (H > 0.51 & H < 0.72 & S > 0.4 & V > 0.87) | (H > 0.561 & H < 0.634 & S > 0.55 & V > 0.5) | (H > 0.555 & H < 0.7 & S > 0.73 & V > 0.2).
   4) *Imdilate* is used to dilate detected areas with disk shape with 2 pixels' radius.
   5) Bwpropfilt is used to choose area between 200 to 8000 pixels for red color, 300 to 10000 for blue color.
   6) Connect close area with center distance less than 75 pixels for red only. Centroid property is used in the *regionprops* function.
   7) Delete all narrow areas detected. MajorAxisLength and MinorAxisLength properties are used in the *regionprops* function.
   8) Create a mask focus only on upper 2/3 of the image in that the traffic sign will never show up in lower 1/3.
   9) Combine the mask for red color, blue color and region of interest. This is the answer for the part 1: Traffic Sign Detection. Toggle to uncomment the line #83 and set a breakpoint in the line #85 to see the results.
2. Traffic sign classification.
   1) *Imresize* all the training images to 64x64 pixels and store them using *imageDatastore*.
   2) *extracHOGFeatures* is used to extract HOF features for every image.
   3) *fitcecoc* is used to build the support vector machine.
   4) Pick all areas from the traffic sign detection and compare them with SVM one by one. *Imresize* first, *extractHOGFeatures* then, *predict* finally to get [predictIndex, score].
   5) Choose only the area with score value larger than -0.002.

6) Choose only the areas with predictIndex leading to 1 of the 8 required signs: 45, 21, 38, 35, 17, 1, 14 and 19.
7) Plot the bounding box of preserved areas after step 5 and 6, using BoundingBox property in the function *regionprops*.

3. Get current frame, record video with 30 framerate.
4. Final Results.
    1) Sign #45, sec 1, 6, 80, 81, 82, total 5:

    2) Sign #21, sec 26, total 1:

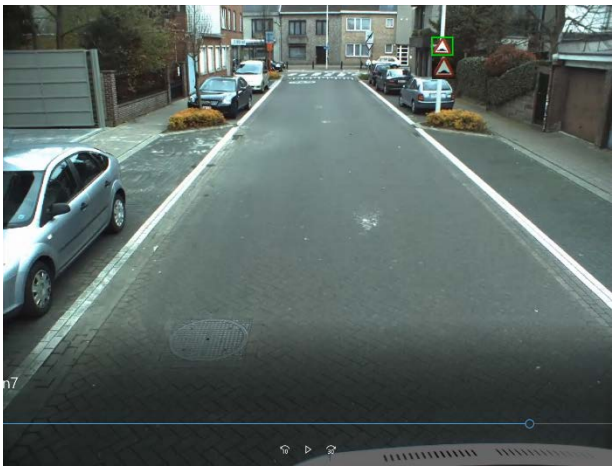


3) Sign #38, sec 26, 36, 49, 74, 83, 92, total 6:

4) Sign #35, sec 33, 36, 48, total 3:

5) Sign #17, sec 31, 35, 42, 46, 86, total 5:

6) Sign #1, sec 72, total 1:



7) Sign #14, sec 67, 70, Total 2:

8) Sign #19, sec 74, 78, total 2:

9) Wrong recognition, sec 24, 28, 77, 81, 90, etc. Mainly comes from the headlights in cars and similar patterns: