Traffic Sign Recognition: Project 4

Explanation & Steps for Traffic Sign Recognition

The following steps were taken into consideration while completing Project-4:

Part – 1: Traffic Sign Classification

1. **Executing the vl_functions**

   From the VLFeat website, the vlfeat-0.9.21-bin.zip file is downloaded and the vl_setup.m file is executed. The code is now ready for execution.

2. **Training of Frames**

   First of all, the frames are fetched from the path and are fed to imageSet to define the collection of images.

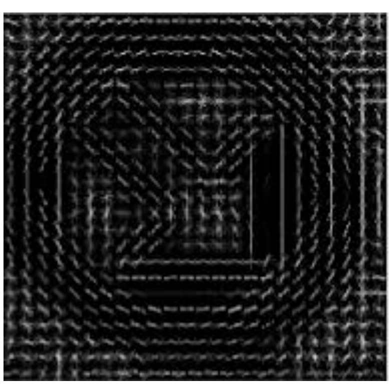
3. **Resizing the Frames**

   As suggested in the question, the frames are resized to 64 x 64 size to easily extract the HOG Features.

4. **Getting the HOG Features**

   Now, using the VLFeat Tutorial as suggested in the question, the HOG parameters are identified. The labels from the training dataset are read thereafter.

   Here, since it is required to specifically detect 8 traffic signs, only those 8 files are kept in the training folder viz. 1, 14, 17, 19, 21, 35, 38 and 45.



5. **Training the Multi-class SVM Model**

   Based on the HOG features, the SVM is trained for various signs (here 8 signs) that would serve as reference for input frames for correctly detecting the signs.

6. **Saving as .mat file**

   To make the future execution easy and independent of the training dataset, the training output is saved as a .mat file to access it while detection of the signs.

The method used here was to find the closest square to the bounding box extracted from the Sign Detection stage and resizing it to a standard size of 64 x 64. HOG features were then extracted and SVM model was used for predicting the signs.

Part-2: Traffic Sign Detection

1. Applying the Gaussian Filter

The imfilter function is used to denoise the frames to remove unwanted noises from the frames.

2. Contrast Normalization

As suggested in the question, with the help of stretchlim function, contrast normalization is carried out to improve the contrast of the image for better performance during its detection.

3. RGB Normalization and Gray scaling of Frame

In order to correctly highlight the blue and red signs, the maximum between the minimum two colors (e.g. R-B and R-G) and 0 is taken which is then converted into grayscale with a threshold of 0.15.

Since here only two colors, blue and red are of primary importance, their addition is done, and that value is then taken in further calculations.

4. Masking the Bottom-Half of Frame

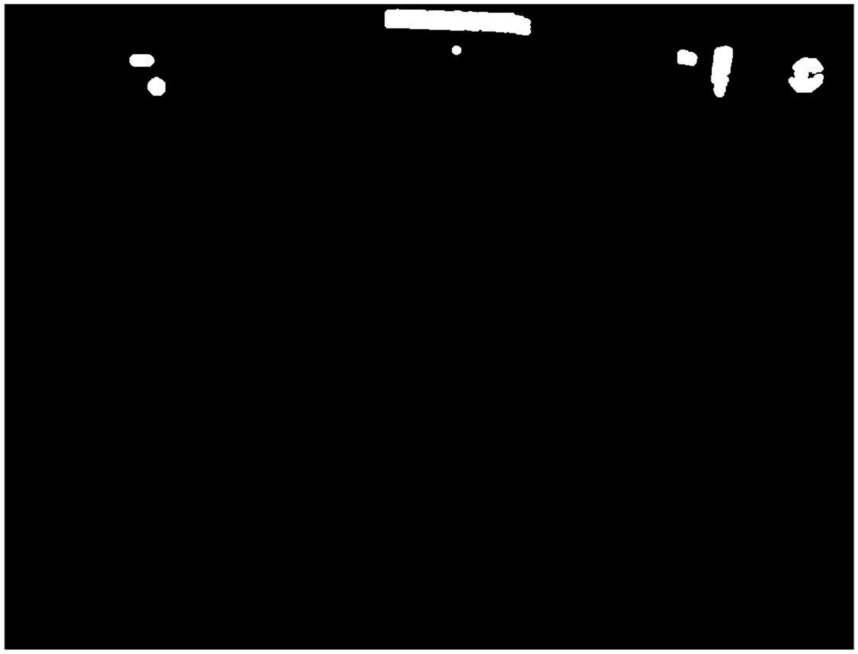
Since, majority of the frames consists the traffic signs in the upper-half of the frame, the bottom portion is masked to avoid detecting unnecessary signs (noises) from cars, poles or hoardings.

5. Implementing the MSER Algorithm

Now, using the vl_mser function used from the VLFeat Tutorial, the value of MinDiversity is set to 0.7, MaxVariation at 0.2 and Delta at 8.

The strel function is then used to represent a flat morphological structural element for morphological dilation and erosion operations.

Here, in order to minimize the false positive regions, the data from the training set has been eliminated to completely avoid false detection of signs.



6. Scaling of Bounding Boxes

The bounding boxes are scaled by taking a ratio to perfectly fit around the entire sign.

7. HOG Features Detection

Now, using the VLFeat Tutorial as suggested in the question, the HOG parameters are identified. The labels from the training dataset are read thereafter.

Here, since it is required to specifically detect 8 traffic signs, only those 8 files are kept in the training folder viz. 1, 14, 17, 19, 21, 35, 38 and 45.

8. Plotting of Bounding Boxes

Based on the training given by SVM and the labels as read from the training dataset, the bounding boxes are plotted around the sign.

Dipam Patel
UID: 115809833

References

1. https://www.mathworks.com/help/images/noise-removal.html
2. https://www.mathworks.com/help/images/ref/stretchlim.html
3. http://benchmark.ini.rub.de/index.php?section=gtsdb&subsection=results&subsubsection=prohibitory
4. https://www.mathworks.com/help/vision/ref/detectmserfeatures.html
5. http://www.vlfeat.org/overview/mser.html
6. http://www.micc.unifi.it/delbimbo/wp-content/uploads/2011/03/slide_corso/A34%20MSER.pdf
7. http://www.vlfeat.org/overview/hog.html
8. https://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/
9. Kishan S Athrey, Bharat M Kambalur, and K Krishna Kumar. Traffic sign recognition using blob analysis and template matching. In *Proceedings of the Sixth International Conference on Computer and Communication Technology 2015*, pages 219–222. ACM, 2015.
10. Saturnino Maldonado-Bascon, Sergio Lafuente-Arroyo, Pedro Gil-Jimenez, Hilario Gomez-Moreno, and Francisco L´opez-Ferreras. Road-sign detection and recognition based on support vector machines. *IEEE transactions on intelligent transportation systems*, 8(2):264–278, 2007.
11. Samuele Salti, Alioscia Petrelli, Federico Tombari, Nicola Fioraio, and Luigi Di Stefano. A traffic sign detection pipeline based on interest region extraction. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–7. IEEE, 2013.
12. Markus Mathias, Radu Timofte, Rodrigo Benenson, and Luc Van Gool. Traffic sign recognition—how far are we from the solution? In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–8. IEEE, 2013.
13. Book on ‘Computer Vision- A Modern Approach’ by Forsyth & Ponce