Introduction to Computer Vision - Traffic Sign Recognition

Redler, Jannes (212923)

jannes.redler@st.ovgu.de

Watson, Oliver (224262)

oliver.watson@st.ovgu.de

Prakash, Subash (220408)

subash.prakash@st.ovgu.de

June 23, 2019

Introduction and Motivation

As mentioned in the paper (Ellahyani et al., 2016), road sign recognition is a challenging problem for many real world applications such as self driving and sign monitoring for maintenance. Factors such as color fading of traffic signs, weather conditions, outdoor lighting and obstacles make the task for recognition more challenging.
The successful detection of traffic signs is a must for the development of autonomous driving. When cars automatically detect signs and change their behaviour as needed, a new era will come. This motivates us with a focus to achieve a high success rate, not just on the given data, but also with other images we took.

Literature Review

There is a good amount of research in the area of traffic sign recognition. The paper (Ellahyani et al., 2016) introduces a new system which involves three stages. At first, segmentation of acquired images to extract ROI (region of interest) using the color information is done. Second, knowing that traffic sign detection is related to geometrical shapes such as circle, triangle, rectangle; a shape classifier is implemented to classify to one of the shapes and only when there is a shape found a traffic sign is recognized on the extracted picture using HOG (Histogram of Oriented Gradients) (Dalal and Triggs, 2005) features. The extracted features are then provided to a Random Forest classifier to perform sign recognition. For our implementation, we have utilized the general idea of this paper.
The paper Dalal and Triggs (2005), introduced Histogram of Oriented Gradients (HOG). The paper utilizes a locally normalized HOG as feature descriptors for Human Detection. They use the

HOG features and perform a classification using SVM (Support Vector Machine). We have used the HOG computation for our implementation and utilized the parameters provided by the paper. The paper (Houben et al., 2013) provides a good overall benchmark approaches used in the area of detection of traffic signs recognition. This helps us with understanding the current state-of-art approaches to such a problem.

All the above works provides us an approach for implementation of road sign recognition.

Implementation

An high level implementation of the project can be viewed from the below diagram:

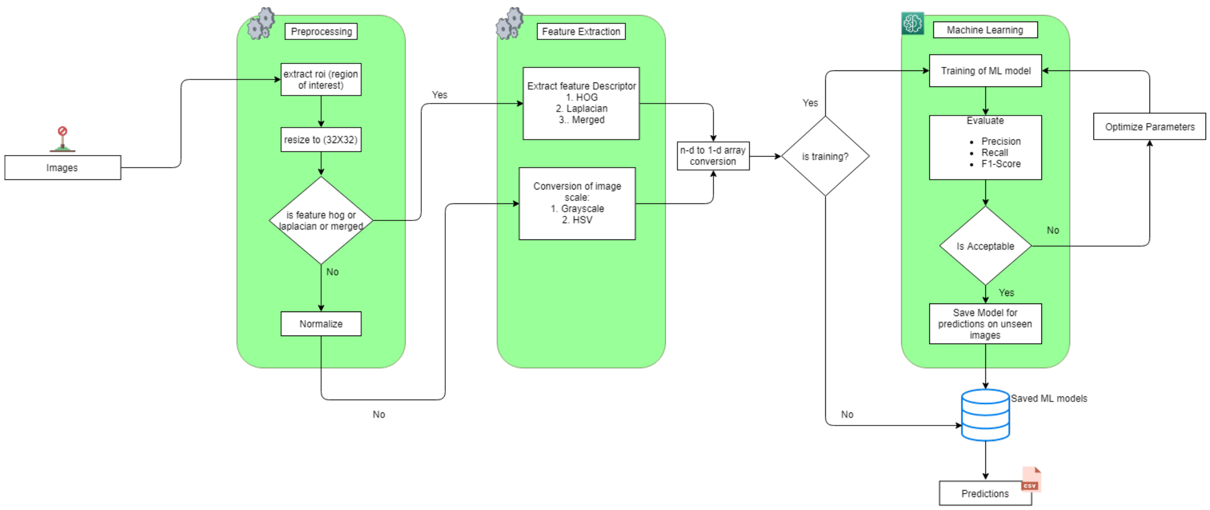

Figure 1: High Level Implementation

Main Methods

The important methods of the project are:

- read_image(base_path=".", roi=False): This function is used to read the images from the base path (image location) and convert them as arrays. There is option to either read only the region of the image (roi) or read the entire image. All the information about filename and region of interest are present in the csv file for each classes. The function will return a list of images and their associated class labels

- extract_features(feature, image_list): Various features such as laplacian, histogram of oriented gradients (hog) are extracted. The parameter of the function is feature which can be ("gray", "hsv", "hog", "laplacian") and the list of images.

  HOG (Histogram of Oriented Gradients): As mentioned in the paper (Dalal and Triggs, 2005) by the authors of the right parameters required for hog. We have utilized the same parameters.
  The image features such as shapes is outlined by using gradients and edges directions.

Steps for computation:

1. **Computing Gradients**: Gradients are computed using Sobel with kernel of (3, 3) across horizontal and vertical direction.
2. **Binning**: A cell is defined. The default parameters used in the paper is (8, 8). It computes a weighted 1-D histogram of gradient.
3. **Creating Descriptor Blocks**: Grouping cells together into a larger block to account for brightness and contrast. Paper suggest to use (16, 16) block size which is 2 times cell size.
4. **Block Normalization**: The block containing group of cells is normalized so that it is not affected by illumination and shadow.
5. **Object recognition**: The obtained features is used to train a machine learning model.

We have utilized opencv HOGDescriptor (OpenCV, 2018) implementation

Laplacian: We furthermore used the Laplace Operator on grayscale images to extract key features.

1. Smooth image with Gaussian
2. Filter with 3x3 Laplacian kernel [0,1,0;1,-4,1;0,1,0]

We have utilized opencv Laplace Operator (OpenCV, 2019) implementation.

- **build_model(X, classId, model_name, features)**: As seen from the figure 1, the feature descriptors are passed on to build a machine learning model. We have written the function with generality in mind, which helps in extending any classifier with minimal changes. For our implementation, we have utilized RandomForest algorithm. We have first utilized a grid search to find the right parameters from the parameters grid.
The parameters utilized for the machine learning system are:

Table 1: Parameter Grid

Parameter	Value
Number Of Trees(n_estimators)	[400, 500, 700]
Criteria of Split	["gini", "entropy"]

After successful search, n_estimator=700, criterion="entropy" was chosen. By utilizing these parameters, a machine learning model is trained for the feature such as hog, laplacian and is saved for future predictions.

- **make_predict(feature)**: This function will make predictions on the unseen test images. Firstly, performs same steps as reading images and extracting features over an unseen images. Secondly, will load the built machine learning model for the provided feature and make prediction on the extracted features

Prediction on a Scene

We have additionally implemented a functionality to make an attempt on predicting traffic signs from the photos we have taken. For this purpose, we have implemented an approach that utilizes template matching to provide the region where the traffic sign might be located in the image. The steps involved in the computation are:

1. **Image threshold**: We first read the image and convert it into HSV format. The next step was to mask the image and perform a bit-wise AND operator on the HSV image. This will remove the unwanted background and extract the required shape. The image below is an example of the process:

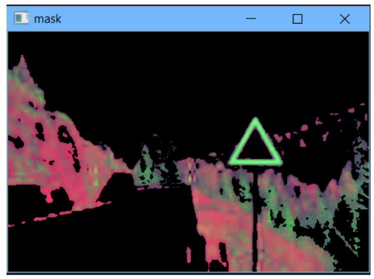

Figure 2: Example of Image Threshold

2. **Template Matching**: A template matching strategy with varying sizes from 50 to 80 is searched to find the region of the traffic sign in the scene. This region is then cropped and prediction is made over an already built machine learning model. The figure 3, show how template matching is powerful enough to identify the traffic sign from a scene.

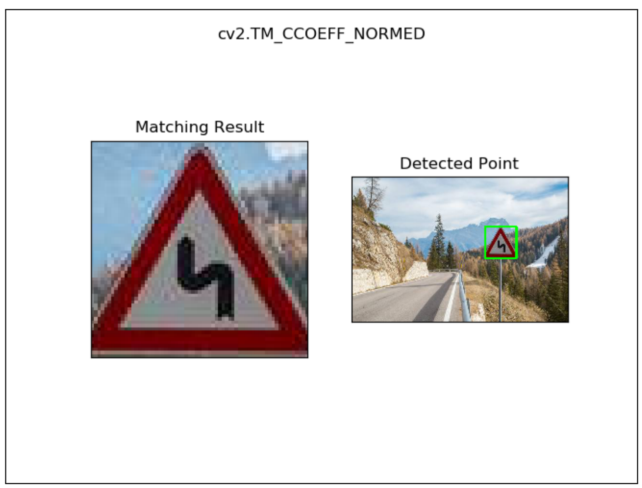

Figure 3: Example of Template Matching result

Evaluation and Results

Training of the different models for each of the features (“gray”,“hsv”,“hog”, “laplacian”) was performed on the German Traffic Sign Recognition Benchmark (GTSRB) (Stallkamp et al., 2011) Training dataset. The data set contains folders containing images of each of the 43 classes. For each class a csv file is also present contatining details of each of the images such as filename, width, height, ROI points and the class ID. Although the task was to train on 4 of the classes from the dataset, we have extended the models and trained them on all 43 classes. After reading all the image files, the dataset was split into "train" and "validation" sets and the models were trained on 2-folds repeated 5 times, producing different splits in each repetition.

The trained models were tested on the GTSRB Test dataset. this contains a single folder with images from all the 43 classes, with a total of 12600 images. The trained models for each of the features were "unpickled" and predictions were performed on this test dataset.

The evaluation measure we have chosen are based on the confusion matrix using notations True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN). The following scores were chosen:

1. **Macro Precision**: Precision is defined as TP / (TP + FP). The macro Precision is the average of the precision of all classes.

2. **Macro Recall**: Recall is defined as TP/ (TP + FN). Similar to the macro Precision, macro Recall is the average of the recall of all classes.

3. **Macro F1 Score**: The macro F1 score is the harmonic mean of macro Precision (MP) and macro Recall (MR) given by (2 * MP * MR) / (MP + MR)

4. **Accuracy**: It is proportion of correctly classified images.

The results of the evaluation are given in table 2 below.

Table 2: Results of the predictions for the different features

Feature / Measure	Grayscale	HSV	HoG	Laplacian	Merged
Macro Precision	0.878643	0.856279	0.929893	0.942971	0.872631
Macro Recall	0.788748	0.751472	0.875729	0.842145	0.790068
Macro F1 Score	0.817345	0.772285	0.894663	0.878313	0.816910
Accuracy	0.864371	0.834283	0.914727	0.915281	0.848852

As we can observe from the the best models are the ones using Laplacian and HoG features. The Laplacian model has a better macro-Precision score and the HoG model has a better macro-Recall score. The Hog model has a better average as indicated by the macro-F1 Score. The Laplacian has a better accuracy of 0.05% to that of the HoG model. Both the Grayscale and the HSV model are considerably worse. The model built from the combined HOG and Laplacian features had scores similar to that of the grayscale model.

The additional traffic sign detection (localization) implementation works very well on scenes that contain only the traffic sign with a simple background with no other high intensity as shown in figure 3. However, detection in scenes with other high intensities are that includes traffic lights for example are not as robust. An example for this can be seen in the figure 3.

Conclusions

In this Traffic Sign Recognition project, we were tasked with building a classifier for the German Traffic Sign Recognition Benchmark (GTSRB) datatset. The approach decided upon was feature extraction from the images and training a machine learning model. We have explored different

Figure 4: A full scene for classification

Figure 5: Full scene images with multiple high intensities

features that could be extracted to be used to train the machine learning model such as Grayscale, HSV, HoG and Laplacian. For the machine learning model a Random Forrest Classifier was used with 700 trees and using entropy as the node split criteria. The models were trained by splitting the Training datatset using k-fold cross validation. The results of the evaluation showed that the best features for this task were Laplacian and HoG features, both having comparable scores while the grayscale and HSV were poor comparetively.

Additionally we have also implemented functionality to make an attempt on predicting traffic signs from a full scene. Image thresholding and template matching was used to localize on the traffic sign from the scene which proved to be quite effective. The localized traffic sign is then be passed to the models for classification.

References

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *international Conference on computer vision & Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE Computer Society.

Ellahyani, A., El Ansari, M., and El Jaafari, I. (2016). Traffic sign detection and recognition based on random forests. *Applied Soft Computing*, 46:805–815.

Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., and Igel, C. (2013). Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *The 2013 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE.

OpenCV (2018). Opencv Implementation Of HOGDescriptors. `https://docs.opencv.org/3.4.1/d5/d33/structcv_1_1HOGDescriptor.html`.

OpenCV (2019). Opencv Implementation Of Laplace Operator. `https://docs.opencv.org/3.4/d5/db5/tutorial_laplace_operator.html`.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. (2011). The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *IEEE International Joint Conference on Neural Networks*, pages 1453–1460.