Traffic Sign Recognition using Multi-Scale Template Matching

Adarsh Kuzhipathalil, Faculty of computer science, OVGU Magdeburg, adarsh.kuzhipathalil@st.ovgu.com

Disha Setlur, Faculty of computer science, OVGU Magdeburg, disha.setlur@st.ovgu.de

Keerthana Chand,Faculty of computer science, OVGU Magdeburg, keertha1@st.ovgu.de

***Abstract*—Commute is a very important aspect in everyday life. Intelligent Transport Systems aims at resolving and optimising traffic congestion. Traffic sign recognition is an important aspect for Intelligent Transport Systems with respect to road safety and also efficient usage of the infrastructure. The task of detecting four classes of traffic signs are accomplished with Multi-scale template matching in our approach.**

Keywords—Template matching, Grey Scale Based template matching

I. Introduction

Traffic sign recognition and classification is one of the former system which was introduced as a part of Advanced Driver Aid Systems (ADAS) which paved way towards autonomous vehicle development. The system detects the road traffic signs using the camera mounted on the windshield or front grille and warns the driver regarding the speed limits, caution signs etc. via the instrument panel.

This system is currently fully mature now and works well for most of the situations. The state of the art systems uses deep learning algorithms to segment localise and classify the traffic signs. Our approach towards the problem is to use classical computer vision technique *Template Matching* to detect the traffic signs.

II. The dataset

A. The German Traffic Sign Recognition Benchmark(GTSRB)

This is a multi class dataset which was released as a part of International Joint Conference on Neural Networks (IJCNN) 2011. The dataset consists of more than 40 classes with over 50000 images. The task as part of the project is to classify 4 classes of traffic signs. The four classes include the "Speed limit 60, Priority Road Sign,Yield to cross-wise traffic Sign, and Priority at the next intersection Sign". The traffic signs are shown in figure 1. The training data for these four classes consist of 6990 images in total. The test data consist of 12630 images in total. The test data consist of test images from all 42 classes of traffic signs. Each class consists of a csv file with ground truth containing the labels and region of interest.

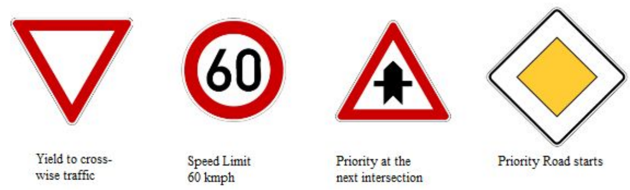

fig 1. The traffic sign classes considered

B.Limitations of the dataset

The dataset consist of images with varying sizes ranging from 25×25 pixels to 150×150 pixels. Accordingly the image quality also varies. A good number of images are really small. The image sizes were studied first in order to understand the dataset. Histogram was plotted to visualise the same. Fig 2 shows a sample histogram of the image sizes of one class. The size distribution is clearly visible that most number of the images are really small as compared to the rest. Same pattern of distribution was observed in all other image classes.

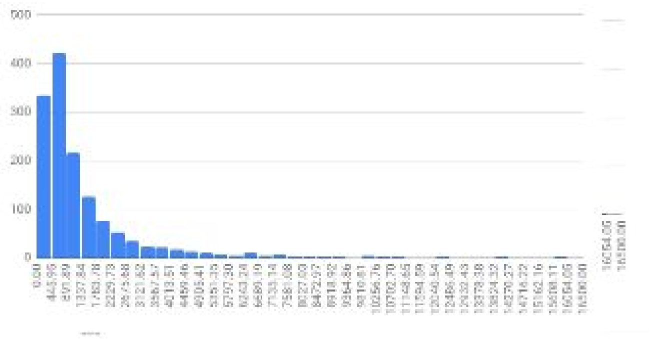

fig 2. histogram of image sizes

Another problem noted with the data is background noise. Some images contain occlusions which cover the traffic signs partly. These images will greatly affect the detection accuracy since the whole traffic sign shape is not seen. Along with noise, illumination differences are another source of error for the data. Some images are perfectly illuminated while the other ones are more or less completely dark. Fig 3 shows some examples of bad data from the Speed limit 60 class.

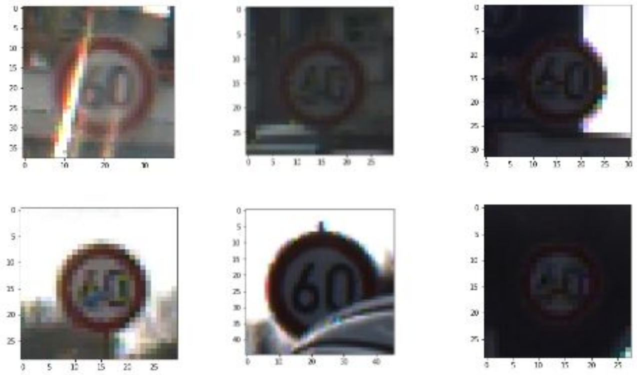

Fig 3. Dataset limitations

III. Method

The template matching technique was decided to be implemented to this dataset as the shapes to be detected are fixed and rigid. There are less variations in the shape and internal features of the image. Since the image size varies greatly scaling was decided to be applied while carrying out the matching process.

Template matching is a technique for finding areas of an image that match (are similar) to a template image (patch). The template slides over the image and match value is calculated at each position. Different matching methods are available to calculate the correlation value between the template and each pixel of the main image. This project uses CCORR_NORMED method for finding the correlation value. This method works as explained in the following equation

$$R(x,y) = \frac{\sum_{x',y'}(T(x',y') \cdot I(x+x',y+y'))}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}}$$

This equation outputs a correlation value within the range 0 to 1 for each pixel of the main image and the template.

Choosing whether the image of the template has to be scaled is a matter of optimising the computational efficiency. Both methods work on the same theoretical principle. But scaling the image and matching with the template has been proved computationally efficient. Hence that method was chosen.

The OpenCV library on python works much efficiently on all aspects of classical and advanced computer vision tasks. The implementation of this project uses OpenCV extensively.

Phase I - Data Preprocessing and Template Selection

The code was developed for loading the images from the dataset folders along with the ground truth from the csv file for the training data. As mentioned earlier the image size study was carried out. This study enabled to decide on the scaling factors of template and images. Since most of the data falls in the category of small images, the decision to concentrate on that region was the outcome of the study.

Phase II - Developing algorithm for Single Scale Template Matching.

The single scale template matching accepts the images in dataset and carries out template matching with the four templates. Thus each image will have four match value matrices. The matrix with maximum match value is considered as the best match. If this match value exceeds the threshold of 0.80, the label of the template is assigned to the image. If the threshold is not matched, it is assigned with a label “0”. The whole algorithm scales the images to 100×100 pixels and matched with the four templates with an average size of 75×75 pixels.

PhaseIII - Extending the algorithm to accommodate multi-scaling of images

The scaling is applied on the images and template matching is carried out.The image is scaled to different sizes and template matching is done with templates. Each pair of template and scaled image set produces a match value matrix. All the match values from the image - template pairs are compared to find the match and classify the traffic sign.

IV. Implementation

The implementation mainly consisted of writing the code and optimising the scaling for the image and template for the training data and then trying the algorithm for test data. The evaluation of the accuracy of the approach is calculated based on the outcome of test data results.

The templates were chosen from the actual traffic signs and masking the background. The Fig 4 shows the chosen templates for the four classes. All the images other than these four are classified as class “0”. Since the test data contains all the classes, the labels other than these four classes is made as “0”

Fig 4. Templates Chosen

Multiple methods were tried to improve the template matching approach

- Gray Scale Based Template Matching

  The image and template are converted into grayscale images and the matching is performed.

The greyscale pixel values of the image is compared with the pixel values of the template

- Template Matching on Binary Images

  The image and templates are converted into binary images using thresholding technique and the matching operation is performed.

- Template matching on Edge detected Images

  Canny edge detector is applied to the images and templates as well. Then matching operation is performed.

- Gaussian blur on the templates and Gray scale images

  The gaussian blur is applied on the templates to smoothen out the solid edges and then applied template matching on the gray scale image.

Out of all the methods which was tried, the Gray Scale Based Template Matching works with the best accuracy.

.

V. Results and Discussions

The algorithm which was optimised for training dataset was applied on the test data directly. Which resulted in an accuracy of 64%. Confusion matrix is plotted with the 4 classes and the fifth data column for the traffic signs undetected class (all classes other than these 4). Fig 5 shows the confusion matrix for the test data.

Classes	All other classes	11	12	13	3
All other classes	7385	67	387	596	1915
11	373	47	0	0	0
12	406	0	265	19	0
13	368	0	0	352	0
3	293	0	0	42	115

Fig 5. Confusion Matrix

The region of confusion matrix marked in red shows the region of the 4 classes which was considered for the project. Most of the misclassifications are happening with the class “0”. Since the threshold which was kept as 0.80, most of the images was classified as undetected . If the threshold was kept low, this will lead to more misclassification into other classes.

Another observation is that more mis classification happens in Class “3”, which is the speed limit 60 class. There are more traffic sign classes in the dataset, and while matching with the template, these traffic signs gave a good match value as the shapes are similar and very few internal features differs for each of these traffic signs.

VI. Scope of Improvement

- Creating templates for all the classes can improve the accuracy greatly. But on the other hand it increases the computational time
- More scaling factors can be added to the algorithm
- Templates with rotation applied to it can also be applied along with the normal templates

VII. Conclusion

Even though the algorithm does not work exceptionally accurate, it works well as far as the computational expense and complexity of the system is concerned. The whole algorithm with two scaling factors and four templates does the matching on the test dataset in 60.41 seconds. That’s a little over a minute for 12630 images on a medium specifications machine. This is good enough to perform the detection in real time with a FPS of 60 or even more.