Intermediary report - traffic sign detection and recognition

1st Mihai Cosmin Cristea
1310A Group
RMA02_01

2nd Andrei Razvan Rautu
1310A Group
RMA02_01

***Abstract*—Driver safety and proper navigation depend on the detection and recognition of traffic signs. In terms of developing technology, intelligent driver aid systems offer a lot of potential. This study proposes an effective algorithm that uses color and shape data to identify traffic signs in photos. The traffic signs are then recognized using auto associative neural networks. This essay focuses on traffic signs with a red background. The system is reliable and capable of detecting traffic signs in unpredictable and complex natural environments. The algorithm is taught to identify the ideal recognition architecture.The experimental findings demonstrate the precision of traffic sign detection and identification.**

Keywords-Traffic sign recognition, Colour Segmentation, Auto associative neural networks(AANN).

I. Introduction

Every year nearly 1.24 million people die in road crashes. There is always a possibility that the driver may not notice the traffic sign and result in road disasters. Also many accidents may occur due to the mood fluctuations of the driver. Thus incorporation of ATSDR system in the vehicle ensures the safety and comfort of drivers by warning them about the presence of traffic signs in different ways. Traffic signs are designed with distinguishable features such as contrast color, shape, and pictogram, so that it can be easily noticed. Many factors influence the visibility and makes it unfavorable for recognition by the system. Prominent among them are [1],

(a) Lighting condition: It varies due to the change in weather conditions (example: rain, twilight, dust, etc.) and also the observed variations throughout the day, especially in the night when the beam light of vehicle falls on traffic sign.

(b) Occlusion: Traffic signs on metropolitan roads are more likely to experience partial occlusion, which changes the contour of the symbol.

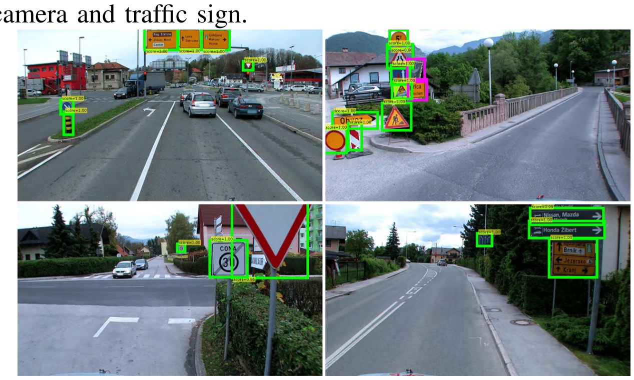
(c) Rotation: Circular shift and distortion in camera projection produces in-plane and out-of-plane rotation respectively.

(d) Multi-class classification: Large number of classes makes classification more challenging.

(e) Cluttered signs and background: The horizontal or vertical arrangement of multiple traffic symbols and presence of unwanted background such as traffic light, static or moving vehicle, car indicator, etc., are to be considered.

(f) Poor image quality: It may be caused due to vibration of vehicle or poor camera quality.

(g) Distance between camera and traffic sign: Traffic sign varies with different scales depending on the distance between camera and traffic sign.



The objective of this work is to show the possibility of video processing algorithm capable of segmenting the traffic signs from images captured and recognise those using auto associative neural networks which is fast, reliable and able to notify drivers about the traffic signs.

This work contains:

1. Introduction
2. State-of-the-Art
3. Related Work
4. Method description
5. Preliminary results
6. Preliminary conclusions

II. State-of-the-Art

The two stages of most traffic-sign recognition algorithms are detection and recognition.

Many studies use thresholding with a specific color space during the segmentation stage to extract the sign color from the image as the initial block of the detection system. Because it is so sensitive to changes in lighting, direct thresholding over the RGB space is rarely employed. As a result, [4] uses a color ratio between the intensity of the individual RGB color components and the sum of RGB intensity to identify strong colors in the image. In [2], one RGB component is used as a reference, which results in a distinct relationship between the RGB components.

Although other areas are utilized, the YUV method is thought to be the most frequently used in [3] to identify blue rectangular signs.

Because color information, which is represented with hue and saturation components, displays small fluctuations for objects of interest with comparable colors, space is the hue saturation intensity (HSI) system. The red and blue colors are extracted in [4] using the correct thresholds on the hue and saturation bands. In [5], two lookup tables are used for each desired color, and a nonlinear transformation over hue and saturation is used to enhance the red and blue hues in the image. [6] calculates a similarity comparison between the hue component and previously saved hue values of specific colors used in traffic signs. This measurement is then passed into a perceptual analyzer that is implemented by a neuronal network.

III. Related-Work

Once they were made available, many standards in the field of driver assistance were universally recognized. The majority of the time, the contribution comprised the definition of relevant evaluation procedures in addition to the publication of original data.

A wide and currently well-studied area of research is traffic sign detection. A thorough review of the most recent changes is provided in the survey of Mgelmose et al. [7].

The two most noticeable characteristics of traffic signs—color and shape—are used in most methods. The management of color is challenging due to the variety of natural lighting situations, and numerous heuristics have been used [8], [9]. Model-based and Viola-Jones-like paradigms can be said to be the two paradigms currently being pursued in the field of form. The model-based systems rely on reliable edge detection and seek to connect irregular polygons or circles to them, typically using a Hough-like voting system or template matching.

The Viola-Jones-like detectors compute a variety of quick and reliable characteristics and attempt to recognize taught patterns using many potentially subpar classifiers.

In addition to our benchmark, there are other publicly accessible datasets that are significant to note here. For example, the Summer Swedish Traffic Signs dataset offers the astounding quantity of 20,000 images from video sequences, 20% of which have been annotated by the authors. Since 2009, the MASTIF project has been putting together data packages with traffic sign sequences that encompass 1,000 to 6,000 photos annually.The stereopolis database includes 273 road signs from 10 different classes and 847 photos total.

These datasets, however, are made up of continuous video sequences that were primarily captured over a single tour and day. As a result, the dataset will contain many instances of the same traffic sign. More significantly, there is little variation in lighting settings and driving situations (country, urban, highway). By offering single photos and showing the majority of traffic sign instances just once, we attempt to address this problem.

IV. Method description

Exploratory data analysis EDA

There is a discrepancy in the expected number of signs since we further expect each sign to be present only at pertinent locations. The 20 kmph sign was the most prevalent sign in this data collection. As the data's incidence rates indicate the historical likelihood of observing a new sign, we won't alter the relative number of these signs. When the model is unable to decide between two indications, keeping the relative ratio of the images constant biases the model toward predicting the more frequent sign.

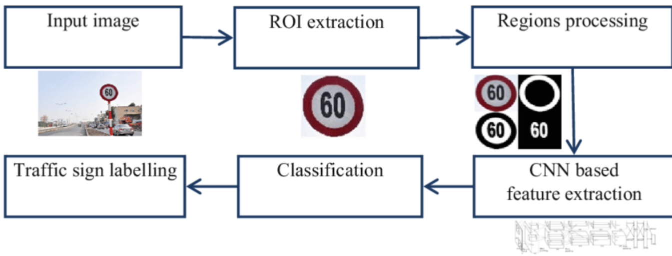
	ClassId	SignName	Occurance
0	2	Speed limit (50km/h)	2250
1	1	Speed limit (30km/h)	2220
2	13	Yield	2160
3	12	Priority road	2100
4	38	Keep right	2070
5	10	No passing for vechiles over 3.5 metric tons	2010
6	4	Speed limit (70km/h)	1980
7	5	Speed limit (80km/h)	1860
8	25	Road work	1500
9	9	No passing	1470

```
data_pd_sorted.tail(10)
```

	ClassId	SignName	Occurance
33	39	Keep left	300
34	29	Bicycles crossing	270
35	24	Road narrows on the right	270
36	41	End of no passing	240
37	42	End of no passing by vechiles over 3.5 metric ...	240
38	32	End of all speed and passing limits	240
39	27	Pedestrians	240
40	37	Go straight or left	210
41	19	Dangerous curve to the left	210
42	0	Speed limit (20km/h)	210

Data augmentation and preprocessing

When using convolutional neural networks (CNNs) for traffic sign detection, there are several important pre-processing and data preparation steps that need to be performed on the dataset. These operations help to ensure that the CNN is able to learn from the data effectively, and can help to improve the accuracy of the model.



Data Collection: The first step is to collect a large dataset of traffic sign images and their corresponding labels. This dataset should include a diverse set of images that represent different traffic signs under different lighting conditions, from various viewpoints, and in varying sizes.

Data Normalization: Once the dataset has been collected, the images need to be normalized to ensure that they are all in the same format. This typically involves resizing the images to a fixed size, such as 32x32 pixels, and converting them to grayscale or RGB format. Normalizing the images helps to make the data more consistent and easier to process.

Data Augmentation: It is also common to apply data augmentation techniques to the dataset in order to increase the amount and diversity of the training data. This can be done by applying various techniques like rotation, translation, flipping, noise addition, brightness, contrast change etc. Data augmentation helps to make the model more robust and generalizable by providing it with more examples of the same traffic sign under different variations.

Data splitting: After data is preprocessed, it is divided into two sets: training set and testing set. It is important to ensure that the data is split in a random and stratified way to ensure that both sets are representative of the entire dataset.

Data Labeling: The images should be labeled with the correct class. For example, an image of a ”stop” sign should be labeled as class 0, an image of a ”speed limit 30” sign should be labeled as class 1, and so on. This is usually done by converting the labels to one-hot encoding.

Handling Imbalanced Data: There might be some classes in the dataset that are under-represented and this might lead to class imbalance problem. This can be handled by oversampling, undersampling or by using the technique called SMOTE (Synthetic Minority Over-sampling Technique).

Deep neural networks may include millions of parameters, making tweaking them difficult without a large amount of data. But sometimes it’s not possible to do that. In these circumstances, data augmentation enables us to provide more training instances.

By performing an affine transformation on the image, we will produce more data samples. Affine transformations, which can be modeled as a linear operation on the matrix, are transformations that do not change the parallelism of lines.

To mimic the effect of observing the sign from various angles and distances, we will explicitly use rotation, shearing, and translation.

MODEL ARCHITECTURE

There are several different algorithm architectures that can be used for traffic sign detection and recognition, each with its own strengths and weaknesses.

Convolutional Neural Networks (CNNs) are a popular choice for traffic sign detection and recognition, due to their ability to learn hierarchical features from images. CNNs typically consist of several layers, including the input layer, convolutional layers, pooling layers, and fully connected layers.

In the input layer, the image is fed into the network. In the convolutional layers, filters are applied to the image to extract features such as edges, corners, and textures. The pooling layers are used to reduce the spatial resolution of the feature maps, and to make the model more robust to small translations of the input image. Fully connected layers are used to learn the final representations of the image.

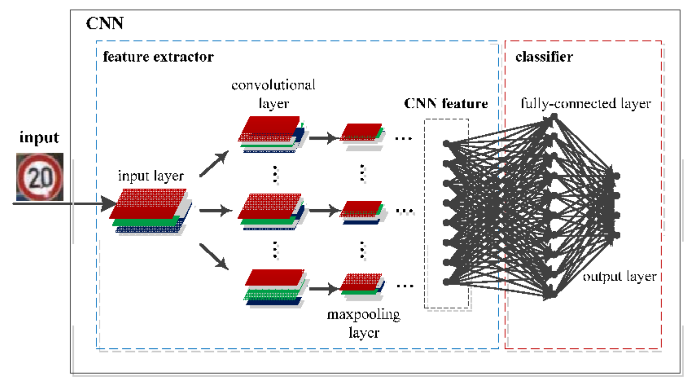


Fig. 1. CNN feature extractor architecture

Another popular architecture for traffic sign detection and recognition is the Multi-task Cascaded Convolutional Networks (MTCNN), which is a three-stage cascaded network that is able to perform face detection and alignment, as well as facial landmark detection. The three stages include a Proposal Network (P-Net), a Refinement Network (R-Net), and an Output Network (O-Net).

The first stage, P-Net, is responsible for generating a set of candidate bounding boxes that may contain a traffic sign. The second stage, R-Net, is used to refine the candidate bounding boxes, and to eliminate false positives. The third stage, O-Net, is used to generate the final bounding boxes and to classify the traffic signs.

Another architecture is the YOLO (You Only Look Once) is a real-time object detection architecture that is able to detect objects in an image with just one pass through the network. It is able to detect multiple traffic signs in a single image, and it is suitable for real-time applications.

Finally, there are architectures based on Region-based CNNs (R-CNNs) like Faster R-CNN, Mask R-CNN and their variants. These architectures are designed for object detection tasks and they propose a region of interest, classify it and then adjust the bounding box to fit the object.

It’s worth noting that these are just a few examples of the many different architectures that can be used for traffic sign detection and recognition, and new architectures are being proposed and developed all the time. The choice of architecture will depend on the specific requirements of the application, such as the desired level of accuracy, real-time performance, and computational resources.

TRAINING

A convolutional neural network (CNN) is typically trained using a process called supervised learning. The basic idea behind supervised learning is to train a model on a labeled dataset, where the model is presented with input data and the corresponding correct output, or label. The goal is to train the model to produce the correct output given a new input that it has not seen before.

Here is an overview of the main steps involved in training a CNN:

Collect a large dataset of labeled images: This dataset should be representative of the problem you are trying to solve, and it should be large enough to provide the model with enough examples to learn from. For example, if you are training a CNN to recognize traffic signs, you would need a dataset of images of traffic signs, labeled with their corresponding class (e.g., ”stop sign,” ”yield sign,” etc.).

Preprocess the data: Before training the model, you may need to preprocess the data to make it suitable for the model. This could include resizing the images to a fixed size, normalizing the pixel values, and converting the images to a format that the model can process.

Define the CNN architecture: This involves choosing the type of layers to use in the network, the number of filters in each layer, the size of the filters, the activation functions, and other parameters. The architecture should be designed to extract useful features from the input images that can be used to distinguish between different classes.

Train the CNN: This involves feeding the input images and their corresponding labels to the model, and adjusting the model’s parameters so that it can produce the correct output for each input. The model is trained using an optimization algorithm, such as stochastic gradient descent (SGD), which minimizes a loss function that measures how well the model is doing at producing the correct output.

Test the CNN: After training the model, it’s important to evaluate its performance on a separate test dataset to check how well it generalizes to new examples. This can be done by comparing the model’s predicted outputs to the true labels for the test examples, and calculating metrics such as accuracy and F1-score.

Fine-tune the model: If the model’s performance is not satisfactory, you can fine-tune it by adjusting the architecture, the parameters, or the preprocessing techniques. You can also try training the model on a larger dataset, or using more advanced techniques such as data augmentation, transfer learning, or ensemble methods.

MODEL PERFORMANCE

The performance of traffic sign detection and recognition models can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score.

Accuracy is the proportion of correctly classified samples to the total number of samples. It is a simple metric to understand, but it can be misleading in situations where the number of samples in different classes is unbalanced.

Precision is the proportion of true positive detections to the total number of positive detections (true positives plus false positives). It measures how many of the detections are actually correct.

Recall is the proportion of true positive detections to the total number of actual positive samples (true positives plus false negatives). It measures how well the model is able to detect all of the actual traffic signs.

F1 score is the harmonic mean of precision and recall, and it is a good overall measure of a model’s performance.

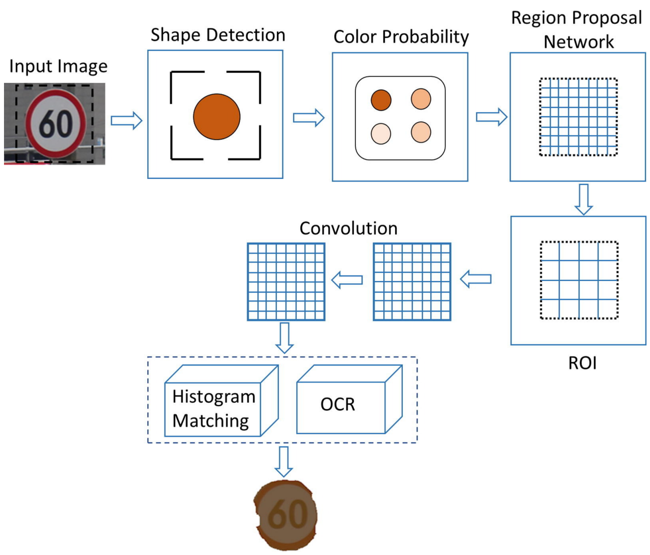
Another metric that is commonly used in object detection tasks is the Intersection over Union (IoU) metric, which is used to evaluate the overlap between the predicted bounding box and the ground-truth bounding box.

Additionally, confusion matrix is often used in this field to analyze the misclassification of certain signs, and to understand the performance of the model in a more detailed way.

It’s also important to note that the performance of traffic sign detection and recognition models can vary depending on the specific dataset and application. For example, the performance of a model trained on European traffic signs may not be as good when applied to traffic signs in Asia, or vice versa. Therefore it’s important to benchmark the model using a diverse set of data and test it in different scenarios to get the most accurate performance measurement.

Also, the performance of the model can be affected by the complexity of the model, the quality of the data and the amount of data used for training and testing. More complex models like CNNs or R-CNNs tend to have better performance than simpler models like SVM, but they may require more computational resources and data.

In general, the performance of traffic sign detection and recognition models is constantly improving as the field of computer vision and machine learning continues to advance.



V. Algorithm Results

Convolutional Neural Networks (CNNs) are a popular choice for traffic sign detection and recognition, due to their ability to learn hierarchical features from images. In recent years, CNN-based models have been achieving state-of-the-art results on various benchmarks for traffic sign detection and recognition.

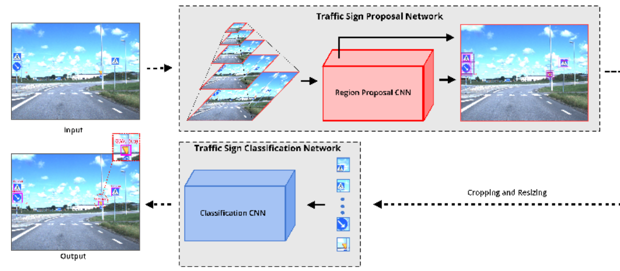
One example of a CNN-based model for traffic sign detection and recognition is the German Traffic Sign Recognition Benchmark (GTSRB) model. This model was trained on a large dataset of German traffic signs and achieved an accuracy of 99.46% on the test set. This model uses a multi-scale approach, where the input image is first resized to different scales and then fed into the CNN. This allows the model to detect traffic signs of different sizes in the image.

Another example is the MultiScale Convolutional Network (MSCNN) model, which was trained on the Belgian Traffic Sign Dataset and achieved an accuracy of 99.14% on the test set. This model uses a multi-scale approach similar to the GTSRB model, but it also incorporates a spatial pyramid pooling layer to extract features at multiple scales.

CNN-based models have also been used in real-time systems, such as driver assistance systems or autonomous vehicles. For example, a CNN-based model was used in a driver assistance system for real-time traffic sign recognition and achieved a detection rate of 98.5% and a recognition rate of 95.4

In recent years, with the advent of deep learning and more powerful computational resources, CNN-based architectures have been improved and evolved, for example using techniques such as:

object detection methods like YOLO, Faster R-CNN, RetinaNet, etc. semantic segmentation methods like FCN, U-Net, etc. Overall, CNN-based models have been achieving very good results on traffic sign detection and recognition task, with high accuracy and real-time performance, making them a viable option for practical applications. However, it's worth noting that the performance of a CNN-based model will depend on the specific dataset and application, and it is important to evaluate the model using a diverse set of data and test it in different scenarios to get the most accurate performance measurement.



VI. Conclusions

Traffic sign detection and recognition is an important technology for modern transportation systems, as it allows vehicles to understand and respond to traffic signs in real-time. There are several different algorithms that can be used for traffic sign detection and recognition, each with its own strengths and weaknesses.

Convolutional Neural Networks (CNNs) are a popular choice for traffic sign detection and recognition, due to their ability to learn hierarchical features from images. CNNs typically consist of several layers, including the input layer, convolutional layers, pooling layers, and fully connected layers.

Other architectures such as Multi-task Cascaded Convolutional Networks (MTCNN) , YOLO and Region-based CNNs (R-CNNs) like Faster R-CNN, Mask R-CNN and their variants, are also popular and effective in object detection tasks, especially when real-time performance is required.

The performance of traffic sign detection and recognition algorithms can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. Additionally, confusion matrix could be used as well to understand the misclassification of certain signs, and to understand the performance of the model in a more detailed way.

Techniques such as data augmentation and transfer learning can be used to improve the performance of these models. However, it's important to note that the performance of traffic sign detection and recognition models can vary depending on the specific dataset and application, and the chosen architecture should take into account the desired level of accuracy, real-time performance, and computational resources.

It's also important to note that traffic sign detection and recognition systems are still facing many challenges like lighting condition, weather, occlusion, and adversarial attacks. Therefore, research in this field is ongoing, and new advancements and improvements are expected in the near future.

References

[1] Gudigar A, Chokkadi S, Raghavendra U (2016) A review on automatic detection and recognition of traffic sign. Multimed Tools Appl 75(1):333–364

[2] H. Kamada, S. Naoi and T. Gotoh, "A compact navigation system using image processing and fuzzy control", Proc. Southeastcon, vol. 1, pp. 337-342, 1990-Apr.

[3] A. de la Escalera, L. E. Moreno, M. A. Salichs and J. M. Armingol, ”Road traffic sign detection and classification”, IEEE Trans. Ind. Electron., vol. 44, no. 6, pp. 848-859, Dec. 1997.

[4] J. Miura, T. Kanda and Y. Shirai, ”An active vision system for real-time traffic sign recognition”, Proc. IEEE Intell. Transp. Syst., pp. 52-57, 2000-Oct.

[5] P. Arnoul, M. Viala, J. Guerin and M. Mergy, ”Traffic signs localisation for highways inventory from a video camera on board a moving collection van”, Proc. IEEE Intell. Veh. Symp., pp. 141-146, 1996-Sep.

[6] A. de la Escalera, J. M. Armingol, J. M. Pastor and F. J. Rodrguez, ”Visual sign information extraction and identification by deformable models for intelligent vehicles”, IEEE Trans. Intell. Transp. Syst., vol. 15, no. 2, pp. 57-68, Jun. 2004.

[7] A. Møgelmose, M. Trivedi, and T. Moeslund, ”Vision based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey, ” IEEE Transactions on Intelligent Transportation Systems, Special Issue on Machine Learning for Traffic Sign Recognition, 2012.

[8] S. Houben, ”A single target voting scheme for traffic sign detection, ” in Proceedings of the IEEE Intelligent Vehicles Symposium, 2011, pp. 124-129.

[9] A. Arlicot, B. Soheilian, and N. Paparoditis, ”Circular road sign extraction from street level images using colour, shape and texture database maps, ” in Proceedings of the Workshop on City Models, Roads and Traffic, 2009.

[10] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[11] Duda RO, Hart PE, Stork DG (2000) Pattern classification, 2nd edn. Wiley-Interscience

[12] Cai D, He X, Zhou K, Han J Bao H (2007) Locality sensitive discriminant analysis. In: Proceedings of the 20th international joint conference on artifical intelligence. Hyderabad, India, pp 708–713

[13] Tenenbaum JB, Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319–2323