TRAFFIC-SIGN RECOGNITION

Marcel PRODAN[1]

Gabriel DOROBANȚU[2]

Narcis IONIȚĂ[3]

Mihai-Lucian VONCILĂ[4]

Nicolae TARBĂ[5]

Costin-Anton BOIANGIU[6]

Nicolae GOGA[7]

Abstract

Traffic-sign recognition is critical for vehicle safety applications, especially as self-driving cars become a reality. This paper proposes a solution based on existing approaches, utilizing deep learning and computer vision preprocessing to create a real-time algorithm that addresses the limitations of previous methods. The proposed algorithm aims to overcome as many drawbacks as possible and serve as a core component of advanced driver-assistance systems (ADAS). The proposed method is evaluated using the German Traffic Sign Recognition Benchmark (GTSRB) and the Belgium Traffic Sign Dataset (BTSD). This study concludes with a fully functional pipeline that can inspire the development of driving assistants and advance the future of self-driving cars.

Keywords: traffic-sign recognition, detection, neural networks, deep learning

JEL Classification: C45, C88

[1] PhD Student, Faculty of Automatic Control and Computers, University "Politehnica" of Bucharest, Romania, e-mail: marcoprod@gmail.com

[2] Student, Faculty of Automatic Control and Computers, University "Politehnica" of Bucharest, Romania, e-mail: gabriel.dorobantu@stud.mec.upb.ro

[3] Student, Faculty of Automatic Control and Computers, University "Politehnica" of Bucharest, Romania, e-mail: narcis.ionita2511@stud.fsa.upb.ro

[4] PhD Student, Faculty of Automatic Control and Computers, University "Politehnica" of Bucharest, Romania, e-mail: mihai_lucian.voncila@stud.acs.upb.ro

[5] PhD Student, Faculty of Automatic Control and Computers, University "Politehnica" of Bucharest, Romania, e-mail: nicolae.tarba@upb.ro

[6] Professor, Faculty of Automatic Control and Computers, University "Politehnica" of Bucharest, Romania, e-mail: costin.boiangiu@cs.pub.ro

[7] Professor, Faculty of Engineering in Foreign Languages, University "Politehnica" of Bucharest, Romania, e-mail: nicu.goga@upb.ro

1. Introduction

According to the World Health Organization's statistic on road traffic injuries [1], 1.3 million people die annually due to road crashes, with 93% of these crashes occurring in less developed countries where older cars with less equipped safety technology are in circulation. Additionally, between 20 to 50 million people get injured each year, leading to a 3% cost of a country's gross domestic product. This highlights the importance of detection and recognition of road signs in the real world. The European New Car Assessment Programme (Euro NCAP) has made it mandatory for all new cars sold in the EU to be equipped with this type of technology soon [2], as they place great value on car safety and have conducted surveys and safety campaigns regarding ADAS, stating that cars of the future need "readable" roads [3]. Detecting different signs in varying weather, daytime and road conditions is seen as a challenge, as current advanced driver assistance systems only have a defined subset of possible signs. Unfortunately, no comprehensive unbiased comparison of sign detecting algorithms has been implemented, and the slow development of this feature may be attributable to a lack of a big freely available benchmark data set.

The recognition process can be divided into two steps: detection and classification. When comparing the two, detection takes precedence because state-of-the-art classification systems can only compete with humans at best. Therefore, classification can be regarded as solved, at least for the time being [4][5]. While most of the attention of sign detection is on particular shapes, such as rectangles and circles, and the type of the sign (speed limit traffic signs), an additional system should be put in place when not focused on a single type of road signs. In most cases, the system uses a color-based segmentation, followed by a recognition stage. However, a large training database with plenty of road signs is required for this approach to work. To minimize the size of the database and get the intended learning process results, the color-based segmentation can be replaced by a combination of color and shape detection.

Key Takeaways:

Global Road Traffic Statistics: Annually, 1.3 million deaths and 20-50 million injuries from road crashes, mostly in less developed countries with older, less safe cars.
Economic Impact: Road accidents cost up to 3% of a country's GDP.
Importance of Road Sign Detection: Essential for reducing accidents; European Union mandates advanced detection technology in new cars.
Challenges in Sign Detection: Varying weather, daylight, road conditions, and limited sign recognition in current systems.
Lack of Comprehensive Studies: No extensive unbiased comparison of sign detection algorithms; hindered by lack of a large, free benchmark dataset.
Recognition Process: Involves detection and classification, with detection being more crucial as classification technology matches human performance.

Focus on Sign Shapes: Current systems primarily recognize shapes like rectangles and circles, specifically for speed limit signs.
Proposed Improvement: Combining color and shape detection for a more effective and compact training database.

2. Related work

Early methods in traffic sign detection utilized a set of rules that im- posed restrictions on color and shape, as well as required signs to appear only in certain regions of an image. These regions are considered candidates, which are then recognized based on a template matching method using other images. Michael Shneier's article [6] on road sign detection is an example of this approach, where his algorithm performed sufficiently to be used and considered in real-time. However, it only addressed warning signs and a few regulatory signs, and its performance suffered when faced with blurry or affected images. Advancements in machine learning have led to the emergence of new approaches in traffic sign recognition. Many articles have proposed different methods that utilize support-vector machines (SVM) or convolutional networks (CNN). For instance, David Soendoro and Iping Supriana's article proposed an SVM method for classifying binary images with localized traffic signs, which resulted from a color-based method with CIELab + hue [7]. A more recent approach used a CNN with fewer parameters, smaller models, and easier training, achieving high accuracy of close to 97%, better than a classical convolutional network [8].

A new and innovative approach to traffic sign recognition is a CNN method that uses GPGPU [9] and Nvidia's latest solution in the automotive industry for autonomous vehicles, called Nvidia DRIVE [10]. This method focuses on solving severe illumination problems regarding low light or wide variance of light-like reflection, in images captured from real-world.

Method	Accuracy
CNN with 3 Spatial Transformers [17]	99.71%
Committee of CNNs [15]	99.46%
COSFIRE filters for object recognition [18]	98.97%
Human Performance [5]	98.84%
Multi-Scale CNNs [16]	98.31%

Method	Accuracy
Random Forests [19]	96.14%
LDA on HOG 2 [5]	95.68%

Table 1. Performance of various methods in the IJCNN2011 Competition

Traditional methods have also made use of a wide range of handcrafted techniques, such as distinctive shapes and colors, such as HOG [11][12] or SIFT [13][14], for classification with machine learning models like SVM, tree classifiers, and boosting. Several traffic sign recognition contests have been established to encourage participation from researchers in a variety of disciplines. The objective of one of these contests, GTSRB, was to produce a report on comparing learning algorithms for traffic sign recognition [5].

By integrating many deep convolutional neural network columns and preprocessing input images into as many little blocks as possible, IDSIA [15] was able to obtain an error rate of 0.54%. COSFIRE, on the other hand, used local and global characteristics with multi-scale CNNs to achieve an error rate of 1.03% [16]. The GTSRB dataset comprises images where the traffic sign takes up a significant amount of the image. However, in the actual world, categorizing images where the traffic sign only takes up a tiny portion of the traffic scene is more significant and should be the primary focus of studies.

Key Takeaways:

Early Traffic Sign Detection Methods: Utilized rule-based systems focusing on color, shape, and specific image regions; relied on template matching for recognition.
Limitations of Early Methods: Targeted mainly warning and regulatory signs; struggled with blurry or affected images.
Advancements with Machine Learning: Introduction of methods using Support-Vector Machines (SVM) and Convolutional Networks (CNN) for improved recognition.
Specific Machine Learning Approaches:
SVM for classifying binary images of localized traffic signs.
CNN with fewer parameters and easier training, achieving up to 97% accuracy.
Innovative Approaches:
CNN methods using GPGPU and Nvidia DRIVE for addressing severe illumination issues in images.
Performance of Various Methods (as per IJCNN2011 Competition):
CNN with Spatial Transformers: 99.71% accuracy.
Committee of CNNs: 99.46% accuracy.

Other methods like COSFIRE filters and Human Performance range between 98.31% to 98.97% accuracy.

Traditional Methods: Employed handcrafted techniques (e.g., HOG, SIFT) combined with machine learning models like SVM for classification.

Traffic Sign Recognition Contests: Encouraged research in diverse disciplines; focused on comparing learning algorithms.

Recent Developments in Deep Learning: IDSIA's integration of multiple CNN columns and preprocessing into small blocks achieved 0.54% error rate; COSFIRE combined local/global characteristics with CNNs for 1.03% error rate.

Significance of Real-World Application: Emphasizes the need for effective categorization in scenarios where traffic signs occupy a small portion of the scene.

3. Beyond State-of-the-Art

Most of the datasets used in state-of-the-art algorithms are focused on one type of sign or have images that heavily emphasize the area in which the sign is located. The goal of this work is to begin by training the proposed algorithm on a traffic sign-focused dataset and then expand it to detect signs in panoramic images. Many challenges associated with traffic sign recognition stem from datasets that are too focused on specific targets rather than real-life situations. Identifying traffic signs in a time-efficient manner is another challenge to be explored, considering that the solution may be implemented on an image capturing device, requiring a timely and relevant outcome. Additionally, the user experience aspect of the problem will be examined. As previously indicated, displaying the result in a non-distracting manner poses a challenge that has yet to be addressed. This entails not only presenting the result appropriately but also implementing a selection process to determine the display worthiness of the data. Although the objectives may seem formidable, the focus is on exploring them and laying the foundation for the next generation of traffic sign recognition software.

Key Takeaways:

The datasets used in current state-of-the-art algorithms typically focus on one type of sign or the sign's location.

The proposed work aims to train an algorithm on a traffic sign-focused dataset, then extend it to panoramic image sign detection.

Existing challenges include datasets being too narrow and not reflecting real-life situations.

Timely and efficient traffic sign recognition is crucial, especially for implementation in image capturing devices.

User experience is a key consideration, particularly in displaying results in a non-distracting manner.

A process for determining the display worthiness of data needs to be established.

The project seeks to explore these challenges and lay groundwork for advanced traffic sign recognition software.

4. Proof of concept

In the initial prototype of the application, a CNN model was developed using the GTSRB dataset [20]. To accomplish this, the Python programming language was utilized along with the NumPy module for mathematical calculations, OpenCV for image processing, and the Tensorflow module and Keras API3 for neural networks and deep learning support. The Scikit-learn library was also applied for streamlined training and testing of the machine learning model.

4.1. Preliminary architecture

The GTSRB dataset is imported and loaded into the application by using the images and their respective labels, which are defined as X and y in the code. The Convolutional Neural Network (CNN) will efficiently process each image because the input images are quite small (resized to 30x30 pixels). The dataset is divided into two models, namely the training model and the test model. The shape of the X train will be *(62734, 30, 30, 3)*, where the first number represents the number of images on which the model is trained. Similarly, the shape of the X test will be *(15684, 30, 30, 3)*, where the first number represents the number of images that are being tested, and the next two numbers represent the size of an image. The last number is the number of color channels, which is 3 for the RGB model.

The model is built using the Keras Sequential function, which allows for the layer-by-layer creation of the model. The layers that will be used are *Conv2D* layers, *MaxPool2D* layers, *Dropout* layers, and *Flatten* layers. *Conv2D* layers use the input images as 2D matrices, *MaxPool2D* layers down-sample the images, Dropout layers ignore random neurons based on a rate to enhance model training, and Flatten layers create a connection between a Conv2D layer and a Dense layer, with the latter being the output layer in the case of neural networks. The activation method used in *Conv2D layers* and some *Dense layers* (except the last one) will be *ReLU*, which stands for Rectified Linear Activation and has been proven to work well in neural networks. The last Dense layer will use the *softmax* activation to transpose the results into probabilities, with 43 nodes, representing each possible class outcome that describes road signs.

The model is built using the *Adam* optimizer to change the learning rate during training. To avoid the manual encoding of the y variable is used the loss parameter sparse categorical cross-entropy. The fit function is used to train the model, where the epochs parameter specifies the number of times the model runs through the data.

To classify a traffic sign, it is necessary to detect the sign first. The application's functionality is illustrated in a pipeline diagram presented in Figure 1.

Figure 1. Pipeline diagram of proposed solution implementation

The proposed scenario is straightforward: a vehicle is equipped with a camera that records the road and captures data for detecting and recognizing traffic signs. The system then displays warnings, alerts, and informative notifications to the driver about their surroundings. Every frame of the video input goes through the detection method, and the regions that are found are then verified before being supplied to the CNN-based recognition classifier. The recognized signs are filtered using the confidence level, and some other parameters are analyzed to determine the appropriate feedback message for the driver. Two approaches were attempted to detect the signs. The first approach used the Fast R-CNN method, which required extensive training time and did not yield conclusive or satisfactory results. The second approach utilized Maximally Stable Extremal Regions (MSER), which required preprocessing of the input image.

Figure 2. First line - blue mask and red mask; Second line - enhanced merged mask and original image with sign detected in a bounding box; In middle - cropped output traffic sign

In the MSER method, detection is split by color to target red or blue traffic signs. To achieve the best results, two separate routines are used for each color. Firstly, to obtain the red mask, the image is processed by performing contrast normalization over each channel and normalizing the red channel intensity. A binary thresholding with a threshold value near the maximum intensity value is then performed to obtain the red mask. Secondly, for the blue mask, contrast enhancement of the original image is done, and it is then converted to the HSV color model for easy segmentation of the blue color. A lower and upper limit of the blue mask is defined to extract the blue area, refer to Figure 2. The resulting red mask and blue mask are merged using a bitwise operation, and the merged mask is dilated to enhance its features. The MSER method is applied to detect the regions of presumed road signs using this mask. Finally, some of the ROIs are discarded by specifying a minimum area, ensuring that the bounding boxes detected are square-like, and checking for any intersecting boxes. If two boxes have a large enough intersection, they are united. The final cropped ROIs are then passed on to the recognition algorithm, as shown in the middle of Figure 2.

4.2. Preliminary results

The model was trained for 3 epochs, and various input and compiling parameters were experimented with. In terms of CNN loss function, sparse categorical cross-entropy has 96.31% accuracy if the color model is RBG and varies to 97.69% for BGR and 97.72% for Grayscale.

For categorical cross-entropy, accuracy varies from 97.19% in RGB to 98.61% for BGR and 99.05% for Grayscale.

The conclusion for this step can be that the best-performing model, which used grayscale input images and categorical cross-entropy loss, achieved an accuracy of 99.05% and took 4 seconds less than the second most accurate model that used OpenCV's BGR model. This result was in line with our expectations based on previous studies, given that the grayscale model has only one-color channel, which explains its faster processing time. The BGR model, surprisingly, performed similarly to the grayscale model, while the RGB model performed the worst.

Based on the preliminary traffic sign detection findings, it can be deduced that additional inputs characterizing the state of the road are necessary for maximum performance under difficult lighting conditions. These inputs can be obtained either by integrating the application with the car's rain and light sensors, or through a Bluetooth connection with the driver's smartphone device that can access the internet and provide weather data. These inputs would be fed into the pipeline along with the camera input, allowing for improved detection of traffic signs in various lighting conditions.

Key Takeaways:

Developed a CNN model for traffic sign recognition using Python, NumPy, OpenCV, TensorFlow, Keras, and Scikit-learn.
Utilized the GTSRB dataset, processing images resized to 30x30 pixels.

The CNN architecture includes Conv2D, MaxPool2D, Dropout, and Flatten layers, with ReLU and softmax activations.
Implemented the Adam optimizer and sparse categorical cross-entropy for training.
Detection of traffic signs is performed first, followed by recognition using the CNN model.
Two detection methods tried: Fast R-CNN (ineffective) and MSER (effective, with color-based segmentation for red and blue signs).
Training the model for 3 epochs showed varying accuracies: highest with grayscale images (99.05%) and categorical cross-entropy loss.
Proposed integrating additional inputs from car sensors or smartphones for improved performance in different lighting conditions.

5. Implementation versus State-of-the-Art

In this section, a comparison will be made between the implementation presented and the results obtained in previous competitions using a given dataset. Specifically, the focus will be on the German Traffic Sign Detection Benchmark and the German Traffic Sign Recognition Benchmark.

5.1. Detection

A benchmark for single-image identification models in the areas of pattern recognition, computer vision, and image-based driver assistance is the German Traffic Sign Detection Benchmark. The dataset consists of 900 photos, separated into three categories that are suitable for multiple detection algorithms with diverse attributes. There are 600 training images and 300 evaluation images.

Algorithm	Prohibitive	Danger	Mandatory
HOG	91.3%	90.7%	69.2%
Hough-like	55.3%	65.1%	34.7%
Voila-Jones	98.8%	74.6%	67.3%

Table 2. The detection rate of all the preliminary algorithms [21]

The Viola-Jones detector, the HOG feature approach, and a model-based technique called Hough-like voting were the first three baseline techniques that were independently trained. Among the pool of independent approaches in the selected category, it was discovered that the Viola-Jones approach had the highest detection rate, with the HOG classifier also performing well. However, for necessary (blue circular) and danger signals, both techniques' performance declined (red triangular). Because they make use of higher-order shape information, the model-based and HOG approaches were better able to manage this challenge.

The German Traffic Sign Detection Benchmark led researchers to the preliminary conclusion that traditional general-purpose detectors worked admirably and outperformed a cutting-edge model-based technique. Performance on unique subsets, like the obligatory signs, was still insufficient for industrial applications. Teams from all around the world responded with innovative solutions after being given a task in response to this.

Among these teams, seven distinguished themselves by achieving perfect results in at least one category. Their methods were more efficient than previous approaches and showed promising results for improving traffic sign detection in challenging scenarios.

Team	Prohibitive	Danger	Mandatory
wgy@HIT501	100%	99.91%	100%
visics	100%	100%	96.98%
LITS1	1000%	98.85%	92%
BolognaCVLab	99.98%	98.72%	95.76%
NII-UIT	98.11%	-	86.97%
wff	-	99.78%	97.62%
milan	-	96.55%	96%

Table 3. Competition Ranking by Area-Under-Curve (Average Overlap) [21]

One issue that is noticeable is how difficult it is to see the required traffic signs. The fact that they are put close to the ground, where they are vulnerable to deterioration or vandalism, and their color - blue, which appear to be difficult to identify in surrounding environment, can also be contributed to this.

Method	Correct recognition rate
Committee of CNNs	99.46%

Human Performance	98.84%
Sermanet	98.31%
Random Forests	96.14%

Table 4. Competition Ranking by Area-Under-Curve (Average Overlap) [21]

5.2. Recognition

The International Joint Conference on Neural Networks hosts the German Traffic Sign Recognition Benchmark, a multi-class, single-image classification task. The dataset is big and is thought to be a lifelike database, with over 40 classifications and over 50,000 photos in total. In the GTSRB final competition stage, four teams stood out from the crowd. Many updated algorithms were presented during the GTSRB competition, and a comparative evaluation of the traffic sign recognition performance of humans and cutting-edge machine learning algorithms was made. The GTSRB concluded that, while the human performance trial achieved a close accuracy of 99.22%, it was exceeded by the effective machine learning strategy - a group of CNNs with a 99.46% right classification rate. Convolutional neural networks, unlike classical computer vision, are capable of learning task-specific features from raw data. Finding the best *ConvNet* architecture for a given task, on the other hand, is mostly empirical.

The proposed approach, which uses categorical cross-entropy loss on a grayscale color model and has a correct recognition rate of 99.05%, is as close to the human performance experiment as the latter got to the machine learning approach. Therefore, the difference between our approach and the human performance experiment is only 0.17%. Comparing these results with those from the GTSRB final ranking would place this solution in second place, 0.21% ahead of third place, and 0.41% below first place.

Key Takeaways:

Comparison of Detection Techniques: The German Traffic Sign Detection Benchmark compared initial techniques like Viola-Jones, HOG, and Hough-like voting for traffic sign detection. Viola-Jones had the highest detection rate, especially for prohibitive signs.
Performance of Detection Algorithms: While traditional general-purpose detectors performed well, their efficacy was limited for certain signs, like mandatory ones. Innovative solutions from international teams showed improvement, with several achieving nearly perfect detection rates in specific categories.
Challenges in Sign Detection: Difficulty in detecting mandatory signs due to factors like placement, deterioration, and color (blue) was noted.

Recognition Benchmark Findings: The German Traffic Sign Recognition Benchmark at the International Joint Conference on Neural Networks highlighted the superiority of machine learning algorithms, particularly Convolutional Neural Networks (CNNs), over human performance in traffic sign recognition.

Algorithmic Advancements: A new approach using categorical cross-entropy loss on a grayscale color model nearly matched human performance in recognition tasks, placing it competitively in the GTSRB final rankings.

Empirical Nature of ConvNet Architecture: Finding the most effective ConvNet architecture for specific tasks remains largely empirical, despite their demonstrated effectiveness in learning task-specific features from raw data.

6. Fine-tuning for the proposed solution

After conducting testing and considering various options, a decision was made to replace the MSER method used for detection. This decision was based on the low accuracy rate of around 60% and the issue of many unwanted regions being identified as candidates due to different lighting scenarios. To address this, the proposed solution utilized a deep neural network (DNN) solution, as it holds great potential for this field. One such system is You Only Look Once (YOLO), a real-time object identification system capable of analyzing images at 30 frames per second (FPS) and even up to 45 frames per second with CUDA acceleration [22]. There are multiple versions of YOLO available, with YOLOv3 being the most suitable for our needs as it prioritizes accuracy over speed compared to YOLOv4. This decision was made with the intention of utilizing GPGPU acceleration later to improve speed of detection.

6.1. Improved architecture

The YOLO system is a real-time object detection solution that is promising for traffic sign detection. After considering different options, the MSER method was replaced due to its low accuracy of detection, around 60%, and the generation of many unwanted regions as candidate objects. The Fast R-CNN also did not prove to be satisfactory, so a deep neural network (DNN) solution was pursued. The YOLOv3 system was found to be the best fit for aimed purpose, with four classes needed for the GTSDB dataset in YOLO format, including prohibitory, danger, mandatory, and other. All 43 subclasses were grouped under each of these classes, from speed limit signs to priorities and directions signs. Pretrained YOLO uses the Darknet architecture and comes with 80 classes from the COCO dataset, but our specific needs required the adjustment to the four classes mentioned above. The model was trained for 8000 iterations, achieving an accuracy of 97.20% after around 8 hours of training, which was a marked improvement over the MSER implementation. However, iterating twice through the dataset for each epoch increased accuracy to 99.11%, though this came at the cost of a longer training time. During detection and recognition, the average data loss was approximately 3.69%, primarily due to different light conditions or camera artifacts such as motion blur and out-of-focus signs. The performance in low light conditions was found to be quite good, with the data loss, in this case, being less significant.

6.2. Unifying models

The initial testing phase involved applying the application on the im- ages from the GTSDB dataset. A unified pipeline was achieved by loading the saved YOLO weights, the YOLO configuration file, and the custom CNN trained model. The YOLO model selects a set of ROIs for each image, and candidates with the best confidence scores and above a threshold are retained for recognition. Based on the maximum probability prediction using softmax, which returns distributed probabilities, a bounding box with the class description and confidence percentage is displayed on the image for the recognized sign. Figure 3 shows the output of the demonstrator application, where traffic signs detected by the YOLO trained model are outlined in a rectangle shape with a text description that consists of the class prediction and its confidence level of the recognition model, displayed in various colors. The detection and recognition rates are optimal when the distance between the road sign and the camera capturing is between 10 to 20 meters. The pipeline can also process a streamed video input in real time for traffic sign recognition. Based on specific parameters, such as display time or accuracy variation, and the output of the recognition or the class type output, different outcome cases can be generated. For example, the time the sign is displayed on the screen of a car, or how or when the driver is notified about the road sign or conditions. A graphical user interface using the Python library tkinter5 can be created for users to test, where based on a video input, a traffic sign is detected and classified to create a notification. Users can provide feedback on which notifications are useful or not and under what circumstances.

Figure 3. Applications detection and recognition results on the GTSDB dataset

Recorded videos from a dashboard camera were used in different scenarios and weather conditions to further test the application. The processes of identification of traffic signs are run in real-time on the videos, not precomputed, to gain experience on the application's behavior, which can be useful for future adjustments.

6.3. Bug tracking and code refactoring

The encountered issues pertained to the misinterpretation of output files and conversion type errors. For instance, during a detection and recognition session, the OpenCV library experienced breakage due to a frame being unreadable or because the last frame was not detected, causing the loop to keep reading none type variables. To address these bugs, new case statements were introduced that either broke the loop or passed to another frame if one existed. Another issue that arose was related to Python's OpenCV library not detecting the CUDA toolkit, which was necessary for obtaining real-time performance for the demonstrator application. Configuring OpenCV, CUDA, and cuDNN (CUDA DNN) for Windows can be quite challenging. The workaround for this problem involved installing the OpenCV library using cmake, which, unfortunately, lacks a precompiled version for Windows, and the compilation itself is a time-consuming task.

Key Takeaways:

Replacement of MSER Method: The MSER method was replaced due to its low accuracy (around 60%) and problems identifying unwanted regions under different lighting conditions.
Adoption of YOLO System: A deep neural network (DNN) solution using YOLO (You Only Look Once) was adopted, capable of processing images at 30-45 FPS with CUDA acceleration.
YOLOv3 Selection: YOLOv3, prioritizing accuracy over speed, was chosen for traffic sign detection, and later enhanced with GPGPU acceleration for faster detection.
Improvements in Architecture: YOLOv3 showed a significant improvement in accuracy (97.20% to 99.11%) for traffic sign detection, with data loss primarily due to lighting and camera artifacts.
Unifying Models for Application: The system unified YOLO weights, configuration, and a custom CNN model, achieving optimal detection and recognition at 10-20 meters distance.
Real-Time Video Processing: The system can process streamed video in real-time for traffic sign recognition, with a user interface for testing and feedback using tkinter5.
Testing with Dashboard Camera: The application was further tested in various weather conditions using dashboard camera videos to observe real-time performance.
Bug Tracking and Refactoring: Addressed issues included misinterpretation of output files and conversion errors, with solutions like introducing new case statements and configuring OpenCV with CUDA for Windows.

8. Conclusions

Precise results were achieved in the recognition stage as shown are presented in the section Related work. The categorical cross-entropy loss function yielded the highest precision, specifically with a value of 99.11% when applied to the grayscale color mode. By comparing this paper's results with those from GTSRB IJCNN as presented in Table 4, it can be concluded that this solution could have been a contender for first place in the competition. The precision achieved was near human precision levels, and any recognition software with over 98% precision is considered suitable for further development and implementation into Level 2+ systems, which do not drive the vehicle but instead provide driver support.

Source	Precision
Best GTSRB Machine Learning Algorithm	99.46%
GTSRB Human Precision	99.22%

Table 5. The final precision comparison

Further development would involve identifying areas where the software performs poorly and devising solutions for these areas, such as accounting for road and weather conditions, as well as driver behavior. To integrate the software into a vehicle, it could be embedded into the automobile's core system with surrounding sensors, or as an accessory such as a Bluetooth dashboard camera that is linked to the driver's mobile phone and serves as a display endpoint.

8.1. Future work

The perfected application will provide options for both online and offline users. The offline version will work seamlessly with the proposed algorithm implementation, while the online version will be served through a web API for handling detection and recognition tasks.

In terms of workflow, the client device will primarily serve as an image capture tool, allowing users to capture images that will be sent to the server via a post request, with the image attached. The server will be responsible for executing the prediction stage, which will culminate in the transmission of a post request back to the client application, containing the prediction results. Depending on the application, the results may be accessed via a dictionary and displayed in various ways, such as using images, sound effects, or other formats.

Key Takeaways:

High Precision Achieved: The research yielded a precision of 99.11% using the categorical cross-entropy loss function in grayscale mode, comparable to top results in the GTSRB IJCNN competition.
Near Human-Level Precision: Achieved precision was close to human levels, indicating suitability for Level 2+ systems offering driver support.
Future Development Focus: Identify and improve areas of software weakness, considering road/weather conditions and driver behavior, with integration options including core system embedding or Bluetooth dashboard cameras linked to mobile phones.
Offline and Online Application Versions: The perfected application will offer both offline (with algorithm implementation) and online (via web API) versions for detection and recognition tasks.
Client-Server Workflow: The client device captures images, sends them to the server, which processes and returns prediction results. The results can be displayed in various formats such as images or sound effects.

References

[1] https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries - Road traffic injuries, 2021, 02.01.2023

[2] https://www.euroncap.com/en/press-media/press-releases/cars-of-the-future-need-readable-roads-now - *Press Release: Cars of the future need "readable" roads now*, 2013, 02.11.2022

[3] Procedure 2018/0145/cod, 2019. 02.11.2022

[4] Jan Salmen Johannes STALLKAMP, Marc SCHLIPSING, Christian IGEL - *Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition* Neural Networks, 32:323–332, 2012

[5] Dan CIREȘAN, Ueli MEIER, Jonathan MASCI, Jürgen SCHMIDHUBER - *A committee of neural networks for traffic sign classification* - 2011 International Joint Conference on Neural Networks, pages 1918–1921, 2011

[6] Michael SHNEIER - *Road sign detection and recognition*, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=823519. 02.11.2022

[7] David SOENDORO, Iping SUPRIANA - *Traffic sign recognition with color-based method, shape-arc estimation and SVM* - Proceedings of the 2011 International Conference on Electrical Engineering and Informatics. IEEE, July 2011.

[8] Wenhui LI, Daihui LI, Shangyou ZENG - *Traffic sign recognition with a small convolutional neural network* - IOP Conference Series: Materials Science and Engineering, 688(4):044034, Nov 2019

[9] Kwangyong LIM, Yongwon HONG, Yeongwoo CHOI, Hyeran BYUN - *Real-time traf- fic sign recognition based on a general purpose GPU and deep-learning* - PLoS One, 12(3):e0173317, March 2017

[10] https://www.nvidia.com/en-us/self-driving-cars - Self-driving cars. 01.11.2022.

[11] Zhiyong HUANG, Yuanlong YU, Jason GU, Huaping LIU - *An efficient method for traffic sign recognition based on extreme learning machine* - IEEE Transactions on Cybernetics, 47(4):920–933, 2017

[12] Yi YANG, Hengliang LUO, Huarong XU, Fuchao WU -*Towards real-time traffic sign detection and classification* - IEEE Transactions on Intelligent Transportation Systems, 17(7):2022–2031, 2016

[13] Fatin ZAKLOUTA, Bogdan STANCIULESCU - *Real-time traffic-sign recognition using tree classifiers* - IEEE Transactions on Intelligent Transportation Systems, 13(4):1507–1514, 2012

[14] Jack GREENHALGH, Majid MIRMEHDI - *Real-time detection and recognition of road traffic signs* - IEEE Transactions on Intelligent Transportation Systems, 13(4):1498–1506, 2012

[15] Dan CIREȘAN, Ueli MEIER, Jonathan MASCI, Jürgen SCHMIDHUBER - *Multi-column deep neural network for traffic sign classification. Neural Networks*, 32:333–338, 2012

[16] Pierre SERMANET, Yann LECUN - *Traffic sign recognition with multi-scale convolutional networks* - 2011 International Joint Conference on Neural Networks, pages 2809–2813, 2011

[17] Álvaro ARCOS-GARCÍA, Juan A. ÁLVAREZ-GARCÍA, Luis M. SORIA-MORILLO - *CNN with 3 spatial transformers*. Neural Networks, 32, 2012

[18] Baris GECER, George AZZOPARDI, Nicolai PETKOV - *Color-blob-based cosfire filters for object recognition* - Image and Vision Computing, 57:165–174, 2017

[19] Fatin ZAKLOUTA, Bogdan STANCIULESCU, Omar HAMDOUN - *Traffic sign classification using k-d trees and random forests* - 2011 International Joint Conference on Neural Networks, pages 2151–2155, 2011

[20] https://benchmark.ini.rub.de - German traffic sign benchmarks, 2019. 10.11.2022

[21] Sebastian HOUBEN, Johannes STALLKAMP, Jan SALMEN, Marc SCHLIPSING, Christian IGEL - *Detection of traffic signs in real-world images: The German traffic sign detection benchmark* - 2013 International Joint Conference on Neural Networks (IJCNN), pages 1–8, 2013

[22] https://pjreddie.com/darknet/yolo - Yolo: Real-time object detection. 26.01.2023.

Bibliography

Álvaro ARCOS-GARCÍA, Juan A. ÁLVAREZ-GARCÍA, Luis M. SORIA-MORILLO - *CNN with 3 spatial transformers*. Neural Networks, 32, 2012

Baris GECER, George AZZOPARDI, Nicolai PETKOV - *Color-blob-based cosfire filters for object recognition* - Image and Vision Computing, 57:165–174, 2017

Dan CIREȘAN, Ueli MEIER, Jonathan MASCI, Jürgen SCHMIDHUBER - *A committee of neural networks for traffic sign classification* - 2011 International Joint Conference on Neural Networks, pages 1918–1921, 2011

Dan CIREȘAN, Ueli MEIER, Jonathan MASCI, Jürgen SCHMIDHUBER - *Multi-column deep neural network for traffic sign classification. Neural Networks*, 32:333–338, 2012

David SOENDORO, Iping SUPRIANA - *Traffic sign recognition with color-based method, shape-arc estimation and SVM* - Proceedings of the 2011 International Conference on Electrical Engineering and Informatics. IEEE, July 2011.

Fatin ZAKLOUTA, Bogdan STANCIULESCU, Omar HAMDOUN - *Traffic sign classification using k-d trees and random forests* - 2011 International Joint Conference on Neural Networks, pages 2151–2155, 2011

Fatin ZAKLOUTA, Bogdan STANCIULESCU - *Real-time traffic-sign recognition using tree classifiers* - IEEE Transactions on Intelligent Transportation Systems, 13(4):1507–1514, 2012

François CHOLLET – *Deep Learning with Python* – Manning Publishing, November 2017, ISBN 9781617294433

Ian GOODFELLOW, Yoshua BENGIO, Aaron COURVILLE - *Deep Learning* – MIT Press, 2016

Jack GREENHALGH, Majid MIRMEHDI - *Real-time detection and recognition of road traffic signs* - IEEE Transactions on Intelligent Transportation Systems, 13(4):1498–1506, 2012

Jan Salmen Johannes STALLKAMP, Marc SCHLIPSING, Christian IGEL - *Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition* Neural Networks, 32:323–332, 2012

Kwangyong LIM, Yongwon HONG, Yeongwoo CHOI, Hyeran BYUN - *Real-time traffic sign recognition based on a general purpose GPU and deep-learning* - PLoS One, 12(3):e0173317, March 2017

Michael SHNEIER - *Road sign detection and recognition*, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=823519. 02.11.2022

Nadra BEN ROMDHANE, Hazar MLIKI, Mohamed HAMMAMI - *An improved traffic signs recognition and tracking method for driver assistance system* - 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, Japan, 2016

Neel BHATT, Pratiksha LALDAS, Vivian Brian LOBO - *A Real-Time Traffic Sign Detection and Recognition System on Hybrid Dataset using CNN* - 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2022

Njayou YOUSSOUF - *Traffic sign classification using CNN and detection using faster-RCNN and YOLOV4* - Heliyon, Volume 8, Issue 12, 2022, ISSN 2405-8440

Pierre SERMANET, Yann LECUN - *Traffic sign recognition with multi-scale convolutional networks* - 2011 International Joint Conference on Neural Networks, pages 2809–2813, 2011

Preeti BAILKE, Kunjal AGRAWAL, *Traffic Sign Classification Using CNN* – IJRASET, ISSN 2321-9653, 2022

Sebastian HOUBEN, Johannes STALLKAMP, Jan SALMEN, Marc SCHLIPSING, Christian IGEL - *Detection of traffic signs in real-world images: The German traffic sign detection benchmark* - 2013 International Joint Conference on Neural Networks (IJCNN), pages 1–8, 2013

Yi YANG, Hengliang LUO, Huarong XU, Fuchao WU -*Towards real-time traffic sign detection and classification* - IEEE Transactions on Intelligent Transportation Systems, 17(7):2022–2031, 2016

Ying SUN, Pingshu GE, Dequan LIU - *Traffic Sign Detection and Recognition Based on Convolutional Neural Network* - 2019 Chinese Automation Congress (CAC), Hangzhou, China, 2019

Wenhui LI, Daihui LI, Shangyou ZENG - *Traffic sign recognition with a small convolutional neural network* - IOP Conference Series: Materials Science and Engineering, 688(4):044034, Nov 2019

Zhiyong HUANG, Yuanlong YU, Jason GU, Huaping LIU - *An efficient method for traffic sign recognition based on extreme learning machine* - IEEE Transactions on Cybernetics, 47(4):920–933, 2017

https://benchmark.ini.rub.de - German traffic sign benchmarks, 2019. 10.11.2022

https://www.euroncap.com/en/press-media/press-releases/cars-of-the-future-need-readable-roads-now - *Press Release: Cars of the future need "readable" roads now*, 2013, 02.11.2022

https://www.nvidia.com/en-us/self-driving-cars - Self-driving cars. 01.11.2022.

https://pjreddie.com/darknet/yolo - Yolo: Real-time object detection. 26.01.2023.

https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries - Road traffic injuries, 2021, 02.01.2023