Traffic Sign Classification for Autonomous Driving

David Akhihiero
Department of Mechanical, Materials and Aerospace Engineering
West Virginia University
West Virginia, U.S.A

***Abstract*—The performance of autonomous vehicles relies on advanced perception systems that can precisely identify and classify traffic signs. This project focused on traffic sign classification while presuming that the detection of traffic signs had already been accomplished. The project implemented a Support Vector Machine (SVM) classifier because it performs well in high-dimensional spaces and is robust to overfitting when working with small datasets. Additionally, a Convolutional Neural Network (CNN) was trained to learn feature representations directly from raw images, leveraging its ability to capture complex patterns for improved classification accuracy. The German Traffic Sign Recognition Benchmark (GTSRB) dataset was used for both training and evaluation. The images from the dataset underwent pre-processing, including resizing and grayscale conversion, to achieve uniformity and improve computational efficiency. Feature extraction methods, including Histogram of Oriented Gradients (HOG) and Scale-Invariant Feature Transform (SIFT), were explored for the SVM classifier. The classifiers' performance was evaluated using accuracy, precision, recall, and F1-score. The outcome was a well-trained SVM and CNN model, allowing for a comparative analysis of their performance in traffic sign classification. The SVM with SIFT features achieved an accuracy of approximately 92%, while the CNN achieved an accuracy of about 95%.**

I. Introduction

The United Nations predicts that road fatalities will increase by up to 50% between the years 2010-2020, amounting to an estimated 1.9 million deaths [1]. In response to this growing trend, the UN launched the "Decade of Action for Road Safety" campaign in 2011. One of the strongest solutions to counter road accidents is by means of Driver Assistance Systems, which automatically manage critical tasks such as lane departure warning and traffic sign recognition.

Traffic sign recognition is currently a fast-expanding component of smart vehicle systems. Traffic signs display critical data, but drivers may overlook or miss them due to factors like fatigue, distraction, or unfavorable weather conditions. Therefore, optimizing road safety through advanced automatic detection and recognition systems is crucial in reducing traffic fatalities. However, these systems face obstacles like varying lighting, diverse weather conditions, occlusions, and rotations that make them perform poorly.

Recognition of traffic signs is important in autonomous vehicles that are also expected to transform transportation to make it safer and efficient. For safe mobility, such vehicles need to understand their environments, including reading and recognizing traffic signs. In the event that a self-driving car reads a sign incorrectly, it can fail to make proper choices, such as stopping at a junction, conforming to speed limits, or

Fig. 1. Image representing typical traffic signs encountered in autonomous driving scenarios.

yielding to pedestrians. Getting the classification for a sign is crucial as misinterpretation can lead to mistakes such as running a red light which can be dangerous. These errors would put both the vehicle's occupants as well as other road users at risk. Similarly, incorrect identification of speed limits results in unsuitable speeds, this is especially problematic in areas with fluctuating speed limits.

Traffic sign recognition systems generally follow three phases: localization, detection, and classification. While detection and recognition of traffic signs at high recall is hard, high precision in real-time tasks, especially with video, is harder. Classical computer vision methods have been employed for traffic sign classification for a long time, dealing with accuracy and efficiency challenges. However, deep learning models, i.e., Convolutional Neural Networks (CNNs) [2], capable of learning complex feature representations end-to-end from raw images are of growing interest to improve classification performance. This project explored both traditional machine learning and deep learning approaches for traffic sign classification; comparing the Support Vector Machines (SVM) [3] with hand-designed features and CNN-based models learned on raw images.

II. Background and Related Works

Traffic sign recognition is an essential component of autonomous driving systems for many years, and in order to be able to drive safely, such systems should be capable of recognizing and reading traffic signs, which offer critical information used in driving decisions. However, traffic sign detection and recognition in real-time, especially in video, is very challenging. Traffic sign recognition generally has three key steps: localization, detection, and classification. Localization is identifying where in an image or video frame traffic signs are, detection is verifying that those locations contain real signs, and classification is identifying what sign it is.

Feature-based techniques had been the norm in traditional computer vision for a long time, employing methods like Histogram of Oriented Gradients (HOG) [4] and Scale-Invariant Feature Transform (SIFT) [5] to extract significant information from images. HOG is a method that captures the edge and shape information of objects and hence is most suitable for detecting the presence of traffic signs based on their shape features. SIFT, on the other hand, finds scale-invariant, rotation-invariant, and affine-invariant key points in an image useful in identifying signs even if they appear in different orientations or at different distances. These features tend to be entered into classifiers like Support Vector Machines (SVMs), which try to label the object by the determined features. Some works including those of [6] have used the HOG method with support vector machines to achieve high classification accuracies. SIFT has also been used in with SVMs in some other works like [7] for traffic sign classification.

While all these techniques hold up well when conditions are ideal, they face tremendous stress as soon as one introduces them in real-world setups. Lighting irregularities, atmospheric conditions, occlusions, and rotations can profoundly degenerate their performance. For instance, an obstructed sign by a moving car or a weather-occluded sign is hard for traditional approaches to detect. Further, hand-crafted feature extraction processes rely on expertise from a specific domain and likely will not extract all the information needed to make accurate classification, particularly in difficult or dynamic conditions.

Deep learning has revolutionized traffic sign recognition over the past several years. Convolutional Neural Networks (CNNs), being one of the deep learning networks, have also been discovered to perform well in tasks like image classification [8]. CNNs can learn automatically hierarchical feature representations from raw images, thus making them easily adaptable to environmental changes without the need for explicit feature engineering. By training on big data sets, CNNs can be made to recognize large numbers of traffic signs under many different conditions and with much higher accuracy than traditional methods.

Even though CNNs have demonstrated high performance, they are not problem-free. One of the greatest problems is that they need large amounts of labeled training data, which at times is not readily available or easy to obtain. Additionally, CNNs generally require a lot of computational power, both during model training and real-time inference in embedded systems. Therefore, most traffic sign recognition systems still rely on a combination of deep learning methods with traditional ones to strike a balance between accuracy and efficiency.

One of the most widely used datasets in traffic sign recognition is the German Traffic Sign Recognition Benchmark (GTSRB) dataset [9]. It includes images of 43 different classes of traffic sign classes and has been extensively utilized to test the performance of a variety of recognition systems. It has been used by several studies to compare traditional machine learning methods, such as SVMs employing HOG and SIFT features, with state-of-the-art CNN-based methods. Results have shown that while CNNs overall perform better than the traditional methods in terms of accuracy, SVMs using manually engineered features like HOG and SIFT still offer competitive results, especially where computational resources are limited.

One of the most significant issues with traffic sign detection, particularly with video streams, is the issue of false positives. Because of the dynamic nature of driving scenes, detection of each traffic sign in every frame of a video will lead to a tremendous number of false alarms when the system is not able to properly eliminate unwanted objects. False alarms are especially problematic in applications such as real-time video analysis, where excessive false detections can lead to system failure or user frustration. Reducing false positives with high recall and accuracy remains a significant challenge in designing robust traffic sign recognition systems.

Researchers have also begun exploring hybrid approaches combining both conventional machine learning techniques and deep learning models in the last few years. For example, some works employed SVMs for initial detection of potential traffic signs with further classification by CNN-based techniques for better identification [10]. This combination has the potential to offer the best of both worlds: high computational performance of conventional techniques and high accuracy of deep learning models.

In summary, while traditional feature-based methods like HOG, SIFT, and SVMs have laid a solid foundation for traffic sign recognition, deep learning models, particularly CNNs, have been more efficient. However, real-time performance, false positives, and the need for large datasets and high computational capabilities are still points of concern. Researchers continue to explore hybrid systems that combine the strengths of both worlds, as well as techniques for optimizing deep learning models for real-time applications.

III. Problem Formulation

The problem is defined as a multi-class classification task where an input image I of a traffic sign is assigned to one of 43 predefined categories from the GTSRB dataset [9]. Given an image I, the objective is to extract meaningful features $F(I)$ and train a classifier C that predicts the correct traffic sign label y:

$$C(F(I)) \to y, \quad y \in \{1, 2, \ldots, 43\}$$

where $F(I)$ represents the feature vector extracted from the image, and C is the classifier that outputs the predicted class label.

To address this problem, three different methods are explored, each leveraging distinct approaches for feature extraction and classification. These methods include:

A. Support Vector Machine (SVM) with HOG Features

The first approach involves using Histogram of Oriented Gradients (HOG) features, a well-known technique for capturing edge and texture information from images. In this approach:

- The input image is first cropped to a Region of Interest (ROI) and resized to a fixed size of 40×40 pixels.
- HOG features are extracted from the image, which captures the gradient structure of the image. Mathematically, the HOG feature extraction process can be represented as:

$$\text{HOG}(I) = \{H_1, H_2, \ldots, H_m\}$$

  where H_i represents the histogram of oriented gradients for the i-th cell in the image, and m is the total number of cells in the image grid.
- These features are then used to train a Support Vector Machine (SVM) classifier S, which assigns the image to one of the 43 traffic sign categories:

$$S(\text{HOG}(I)) \to y, \quad y \in \{1, 2, \ldots, 43\}$$

- The trained SVM model is saved for further use in the classification task.

B. Support Vector Machine (SVM) with SIFT Features

The second method utilizes Scale-Invariant Feature Transform (SIFT) features, which are robust to changes in scale, rotation, and affine transformations. In this approach:

- The image is preprocessed by cropping and resizing to the same fixed size of 40×40.
- SIFT keypoints are detected, and descriptors for these keypoints are computed. Let $\{k_1, k_2, \ldots, k_n\}$ represent the set of detected keypoints, and $\{d_1, d_2, \ldots, d_n\}$ represent the corresponding descriptors.
- The descriptors are averaged to form a single feature vector F_{SIFT} representing the image:

$$F_{\text{SIFT}}(I) = \frac{1}{n} \sum_{i=1}^{n} d_i$$

  where n is the number of detected keypoints.
- This feature vector is used to train an SVM classifier S for predicting the traffic sign class:

$$S(F_{\text{SIFT}}(I)) \to y, \quad y \in \{1, 2, \ldots, 43\}$$

- The trained SVM model is saved for future classification tasks.

C. Convolutional Neural Network (CNN)

The third method uses a Convolutional Neural Network (CNN), a deep learning model capable of automatically learning features from raw image data. This method eliminates the need for manual feature extraction and directly learns optimal features for classification. The CNN is structured as follows:

- The network comprises three convolutional blocks C_1, C_2, C_3, each consisting of a 2D convolutional layer with increasing filter depth ($8 \to 16 \to 32$), 'same' padding to preserve spatial resolution, followed by batch normalization and ReLU activation layers. Max-pooling is applied after the first and second convolutional blocks to downsample feature maps and reduce spatial dimensions. For the i-th convolutional layer, the output feature map F_i is computed as:

$$F_i = \text{ReLU}(\text{Conv}(F_{i-1}, W_i) + b_i)$$

  where W_i and b_i are the weights and bias terms for the i-th layer, and F_{i-1} is the input from the previous layer.
- After the convolutional blocks, the feature maps are flattened and passed to a fully connected (dense) layer with 43 output neurons, each corresponding to a traffic sign class. This is followed by a softmax activation layer:

$$S_{\text{softmax}}(F_{\text{CNN}}) = \frac{\exp(F_{\text{CNN}})}{\sum_{j=1}^{43} \exp(F_{\text{CNN},j})}$$

  where F_{CNN} is the feature vector from the last hidden layer and $F_{\text{CNN},j}$ is the logit for class j.
- The network is trained using the Adam optimizer with an initial learning rate $\alpha = 1 \times 10^{-3}$, a maximum of 20 training epochs, and a mini-batch size of 128. The data is shuffled every epoch, and validation occurs periodically during training.
- The trained CNN model is saved and evaluated on the test set to assess its classification performance. The full CNN architecture is visualized in Figure 2.

D. Comparison of Methods

Each of these methods has distinct advantages:

- The SVM with HOG method relies on handcrafted features and is effective for detecting simple textures and edges in images. However, it may be less robust to transformations such as scale changes and rotations.
- The SVM with SIFT method captures more robust features, particularly useful for handling images with scale and rotation variations.
- The CNN approach uses deep learning to automatically extract features, making it highly adaptable and capable of learning complex patterns from raw image data, though it requires a larger dataset and more computational power.

By exploring and comparing these methods, the goal is to identify the most effective approach for real-time traffic sign recognition, suitable for deployment in autonomous vehicles or other real-world applications.

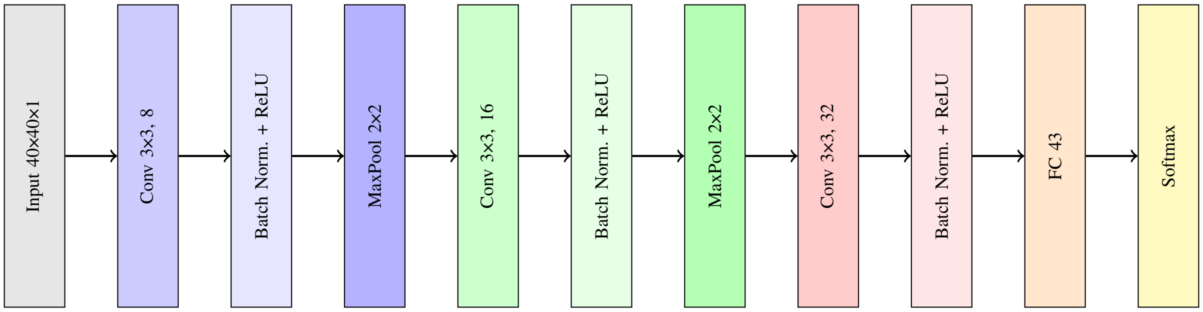

Fig. 2. Visual representation of the CNN architecture for traffic sign classification.

IV. Results and Discussion

The models were evaluated on the German Traffic Sign Recognition Benchmark (GTSRB) dataset [9] consisting of 39,209 training images and 12,630 testing images. Three different approaches were compared: Support Vector Machine (SVM) with HOG features, SVM with SIFT features, and a Convolutional Neural Network (CNN).

To provide a visual representation of the classification task, Figure 3, Figure 4, and Figure 5 show examples of traffic signs with extracted HOG and SIFT features respectively, as well as a complete palette of the 43 traffic sign classes.

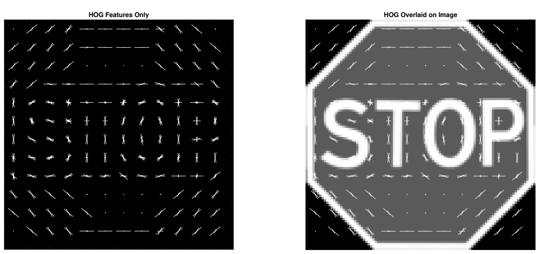

Fig. 3. Traffic STOP sign with HOG features visualized

Fig. 4. Traffic STOP sign with SIFT keypoints and descriptors visualized

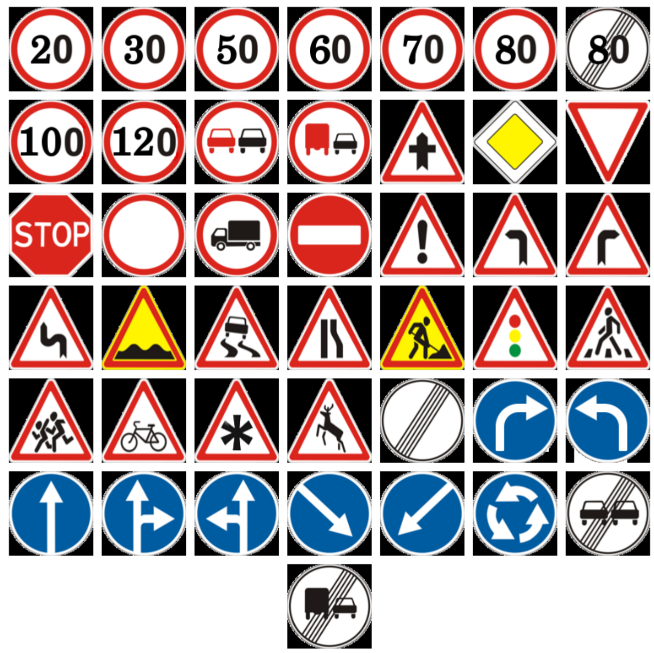

Fig. 5. Palette of all 43 traffic sign classes from the GTSRB dataset

Tables I, II lists the number of training samples for each class, including the traffic sign description and corresponding sample image.

Table IV shows the macro average results for each model and Table IV presents the precision, recall, and F1-scores for each class and model. The CNN model outperformed both SVM variants across all metrics, achieving an overall accuracy of 95.46%. The SVM with HOG features followed with 92.20% accuracy, while SVM with SIFT features yielded a significantly lower accuracy of 51.43%. An interesting trend observed in Table IV is the disparity in execution times across models. Despite their reputation for being computationally expensive, the CNN completed both training and testing significantly faster than the SVM-based models. The CNN achieved a testing time of 67.68 seconds, compared to 1924.51 seconds for the HOG-based SVM and 1859.10 seconds for the SIFT-based SVM. Notably, the SIFT-based SVM took considerably longer to train than the HOG variant, likely

TABLE I
TRAINING SET DISTRIBUTION WITH TRAFFIC SIGN DESCRIPTIONS AND EXAMPLE IMAGES

Class ID	Traffic Sign Description	Training Count	Example Image
0	Speed limit (20km/h)	210	
1	Speed limit (30km/h)	2220	
2	Speed limit (50km/h)	2250	
3	Speed limit (60km/h)	1410	
4	Speed limit (70km/h)	1980	
5	Speed limit (80km/h)	1860	
6	End of speed limit (80km/h)	420	
7	Speed limit (100km/h)	1440	
8	Speed limit (120km/h)	1410	
9	No passing	1470	
10	No passing for vehicles over 3.5 metric tons	2010	
11	Right-of-way at the next intersection	1320	
12	Priority road	2100	
13	Yield	2160	
14	Stop	780	
15	No vehicles	630	
16	Vehicles over 3.5 metric tons prohibited	420	
17	No entry	1110	
18	General caution	1200	
19	Dangerous curve to the left	210	
20	Dangerous curve to the right	360	
21	Double curve	330	
22	Bumpy road	390	
23	Slippery road	510	
24	Road narrows on the right	270	
25	Road work	1500	
26	Traffic signals	600	
27	Pedestrians	240	
28	Children crossing	540	
29	Bicycles crossing	270	
30	Beware of ice/snow	450	

TABLE II
(CONTINUED) TRAINING SET DISTRIBUTION

Class ID	Traffic Sign Description	Training Count	Example Image
31	Wild animals crossing	780	
32	End of all speed and passing limits	240	
33	Turn right ahead	689	
34	Turn left ahead	420	
35	Ahead only	1200	
36	Go straight or right	390	
37	Go straight or left	210	
38	Keep right	2070	
39	Keep left	300	
40	Roundabout mandatory	360	
41	End of no passing	240	
42	End of no passing by vehicles over 3.5 metric tons	240	

due to the complexity of extracting and processing SIFT descriptors and the dimensionality of the resulting feature vectors. This highlights that hand-crafted features, especially SIFT, can introduce significant computational overhead during both training and inference. In contrast, the CNN's end-to-end architecture eliminates the need for separate feature extraction, leading to faster and more efficient model execution. These findings underscore the practical advantage of deep learning models not only in accuracy but also in real-world efficiency, making them more suitable for deployment in autonomous driving systems where time and responsiveness are critical.

TABLE III
MACRO-AVERAGE METRICS AND TESTING TIME FOR EACH CLASSIFICATION MODEL

Model	Accuracy (%)	Macro Precision	Macro Recall	Macro F1-Score	Training Time(s)	Testing Time(s)
SVM with HOG	92.20	0.87	0.94	0.90	1186.06	1924.51
SVM with SIFT	51.43	0.52	0.56	0.53	2777.99	1859.10
CNN	**95.46**	**0.93**	**0.95**	**0.94**	**210.94**	**67.68**

Incorporating class-level images allows for deeper insight into why some models may struggle with certain signs. For instance, rare signs with only a few hundred samples (e.g., Class 0, Class 19, Class 27, Class 37, Class 41, and Class 42) may not offer enough variability during training, impacting model generalization. Similarly, signs with similar visual features (like triangular warning signs) can cause confusion in models with weaker feature discriminability, particularly the SVM with SIFT features.

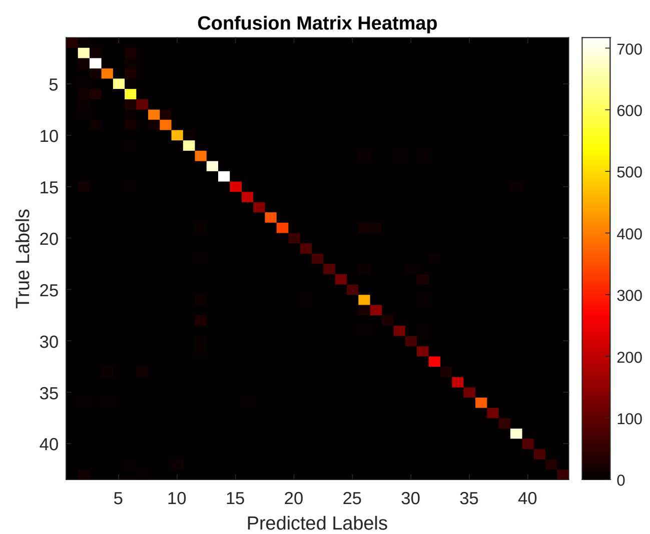

Fig. 6. Confusion Matrix for SVM with HOG features

Figure 6, Figure 7, and Figure 8 display the confusion matrices for each model, respectively. As observed, the CNN confusion matrix shows tighter diagonal alignment, indicating more correct predictions across all classes compared to the SVM-based models. The SVM-SIFT model displays notable dispersion, confirming its poor performance as highlighted in the class-specific metrics.

These findings confirm the efficacy of deep learning methods for visual recognition in autonomous systems, where

TABLE IV
CLASS-SPECIFIC PRECISION, RECALL, AND F1-SCORE FOR SVM WITH HOG, SIFT, AND CNN

Class	**Precision**			**Recall**			**F1-Score**		
	SVM-HOG	SVM-SIFT	CNN	SVM-HOG	SVM-SIFT	CNN	SVM-HOG	SVM-SIFT	CNN
Class 0	0.63	0.60	**0.85**	**1.00**	0.55	0.98	0.78	0.57	**0.91**
Class 1	0.92	0.61	**0.97**	0.84	0.67	**0.97**	0.88	0.64	**0.97**
Class 2	0.95	0.61	**0.99**	0.88	0.22	**0.92**	0.92	0.32	**0.96**
Class 3	0.88	0.39	**0.91**	0.91	0.43	**0.92**	0.89	0.41	**0.91**
Class 4	0.96	0.36	**0.97**	**0.97**	0.54	**0.97**	0.96	0.43	**0.97**
Class 5	0.91	0.42	**0.93**	0.76	0.38	**0.90**	0.83	0.40	**0.91**
Class 6	0.69	0.21	**0.79**	0.81	0.57	**0.98**	0.75	0.30	**0.87**
Class 7	0.88	0.58	**0.96**	0.94	0.60	**0.95**	0.91	0.59	**0.95**
Class 8	0.85	0.38	**0.94**	0.91	0.45	**0.94**	0.88	0.41	**0.94**
Class 9	0.96	0.45	**0.99**	**0.95**	0.63	0.93	**0.96**	0.52	**0.96**
Class 10	**0.98**	0.48	**0.98**	0.96	0.65	**0.98**	0.97	0.55	**0.98**
Class 11	**0.93**	0.53	**0.93**	0.81	0.73	**0.97**	0.87	0.62	**0.95**
Class 12	**1.00**	0.38	0.97	**1.00**	0.61	0.99	**1.00**	0.47	0.98
Class 13	**1.00**	0.63	**1.00**	**1.00**	0.54	0.97	**1.00**	0.58	0.99
Class 14	0.86	0.46	**0.99**	0.98	0.59	**0.99**	0.92	0.52	**0.99**
Class 15	0.98	0.68	**0.99**	0.94	0.73	**0.96**	0.96	0.70	**0.97**
Class 16	0.96	0.73	**0.98**	**1.00**	0.90	**1.00**	0.98	0.81	**0.99**
Class 17	**0.98**	0.60	0.96	**1.00**	0.68	**1.00**	**0.99**	0.64	0.98
Class 18	0.86	0.47	**0.92**	**0.97**	0.51	0.94	0.91	0.49	**0.93**
Class 19	0.98	0.48	**1.00**	**1.00**	0.60	0.87	**0.99**	0.54	0.93
Class 20	**0.97**	0.61	**0.97**	**0.90**	0.50	0.78	**0.93**	0.55	0.86
Class 21	0.82	0.52	**0.91**	0.95	0.60	**0.98**	0.88	0.56	**0.94**
Class 22	0.72	0.62	**0.86**	**1.00**	0.58	0.96	0.84	0.60	**0.91**
Class 23	0.80	0.37	**0.89**	**0.94**	0.32	0.92	0.87	0.35	**0.90**
Class 24	0.90	0.24	**0.93**	**0.99**	0.40	0.92	**0.94**	0.30	0.93
Class 25	0.94	0.67	**0.98**	0.84	0.54	**0.96**	0.88	0.60	**0.97**
Class 26	0.82	0.66	**0.86**	0.84	0.67	**0.87**	0.83	0.66	**0.86**
Class 27	0.47	0.32	**1.00**	0.90	0.36	**0.91**	0.62	0.34	**0.95**
Class 28	0.84	0.51	**0.93**	0.93	0.39	**0.95**	0.88	0.44	**0.94**
Class 29	0.79	0.26	**1.00**	0.86	0.43	**0.99**	0.82	0.32	**0.99**
Class 30	**0.86**	0.23	0.72	0.66	0.35	**0.93**	0.75	0.28	**0.81**
Class 31	**0.97**	0.61	**0.97**	**0.96**	0.75	0.93	**0.97**	0.67	0.95
Class 32	0.50	0.65	**1.00**	**1.00**	0.49	0.95	0.67	0.56	**0.98**
Class 33	**0.99**	0.79	**0.99**	**1.00**	0.74	0.97	**0.99**	0.77	0.98
Class 34	**0.98**	0.89	**0.98**	**0.98**	0.80	0.94	**0.98**	0.84	0.96
Class 35	0.94	0.26	**0.99**	0.97	0.66	**0.98**	0.95	0.38	**0.98**
Class 36	**0.99**	0.61	0.96	**1.00**	0.67	0.98	**1.00**	0.64	0.97
Class 37	**0.90**	0.57	**0.90**	**1.00**	0.49	0.96	**0.95**	0.53	0.93
Class 38	**0.99**	0.64	0.98	0.98	0.59	**0.99**	**0.99**	0.61	0.98
Class 39	**0.98**	0.08	0.94	**0.99**	0.21	0.97	**0.98**	0.11	0.96
Class 40	**0.90**	0.84	0.82	0.98	0.72	**0.99**	**0.94**	0.78	0.90
Class 41	0.58	**0.87**	0.62	**1.00**	0.73	0.77	0.74	**0.79**	0.69
Class 42	0.66	0.34	**0.88**	**0.98**	0.53	**0.98**	0.79	0.42	**0.92**

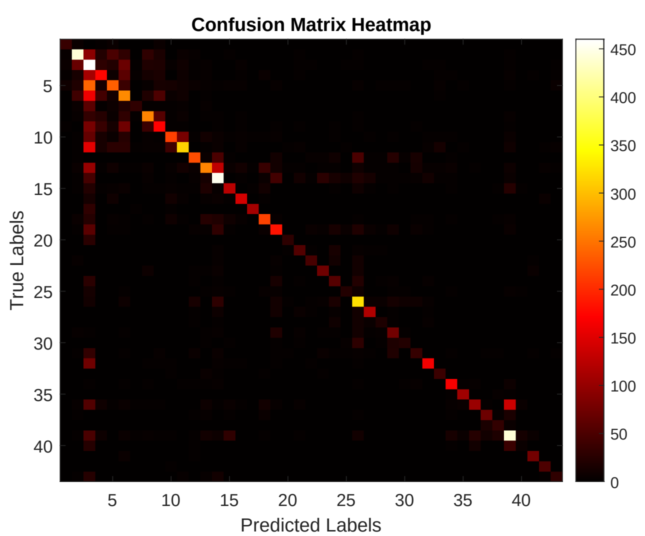

Fig. 7. Confusion Matrix for SVM with SIFT features

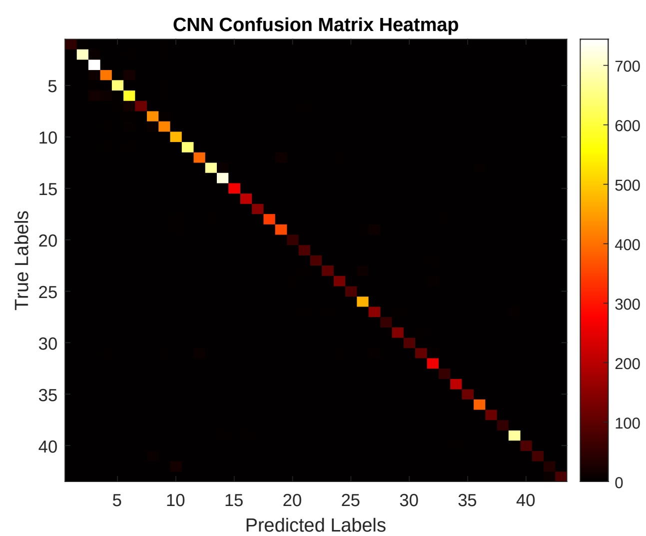

Fig. 8. Confusion Matrix for CNN classifier

generalization and adaptability to real-world variability are paramount.

V. Conclusion

This project compared three traffic sign classification techniques using the German Traffic Sign Recognition Benchmark (GTSRB) dataset [9]: Support Vector Machine (SVM) classifiers using Histogram of Oriented Gradients (HOG) and Scale-Invariant Feature Transform (SIFT) features, and a Convolutional Neural Network (CNN). Of the three methods, the CNN gave the highest classification accuracy (95.46%), which was better than SVM with HOG (92.20%) and SVM with SIFT (51.43%). Surprisingly, the CNN also had the fastest training and testing time, despite the fact that deep learning models tend to be more computationally costly. This is probably due to the fact that extraction of HOG features takes time, and SVM inference using high-dimensional features is comparatively costly. In contrast to this, following training, the CNNs benefits from faster end-to-end inference.

These results show the strength of CNNs in performance as well as in efficiency for deployment in practice. While older approaches like HOG+SVM are still applicable in low-data or no-GPU scenarios, they may incur higher inference costs. Future work could explore lean CNN models, pruning, or hybrid pipelines to meet speed vs. accuracy trade-offs for application in resource-constrained autonomous systems.

References

[1] H. E. Rosen, I. Bari, N. Paichadze, M. Peden, M. Khayesi, J. Monclús, and A. A. Hyder, "Global road safety 2010–18: an analysis of global status reports," *Injury*, 2022.

[2] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 12, pp. 6999–7019, 2021.

[3] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[4] A. Jain and D. Singh, "A review on histogram of oriented gradient," *IITM Journal of Management and IT*, vol. 10, no. 1, pp. 34–36, 2019.

[5] W. Burger and M. J. Burge, "Scale-invariant feature transform (SIFT)," in *Digital Image Processing: An Algorithmic Introduction*. Springer, 2022, pp. 709–763.

[6] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang, "A hierarchical method for traffic sign classification with support vector machines," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2013, pp. 1–6.

[7] X. HUa, X. ZHUa, D. LIa, and H. Li, "Traffic sign recognition using scale invariant feature transform and svm," in *A special joint symposium of ISPRS technical commission IV and AutoCarto in conjunction with ASPRS/CaGIS fall specialty conference November*, 2010, pp. 15–19.

[8] A. Sangal *et al.*, "Traffic signs classification using convolutional neural networks: a review," in *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*. IEEE, 2021, pp. 450–455.

[9] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German traffic sign recognition benchmark: a multi-class classification competition," in *The 2011 international joint conference on neural networks*. IEEE, 2011, pp. 1453–1460.

[10] Y. Sun and L. Chen, "Traffic sign recognition based on CNN and twin support vector machine hybrid model," *Journal of applied mathematics and physics*, vol. 9, no. 12, pp. 3122–3142, 2021.