Global Journal of Engineering Innovations & Interdisciplinary Research

Original Article

Deep Learning Approaches For Traffic Sign Detection

S Ramana Reddy[1], Y Raj Kumar[2], T Sai Kiran[2], T Tarun[2]

[1]Assitant Professor, Department of Artificial Intelligence and Data Science, Vignan Institute of Technology and Science, Hyderabad, India

[2]UG Student, Department of AI&DS, Vignan Institute of Technology and Science, Hyderabad, India

Correspondence

S. Ramana Reddy

Assistant Professor, Department of Artificial Intelligence and Data Science, Vignan Institute of Technology and Science, Hyderabad, India

- Received Date: 25 May 2025
- Accepted Date: 15 June 2025
- Publication Date: 27 June 2025

Keywords

Recognition of traffic signs, detection models, adaptive learning, pre-trained model, detection models, Resnet, Inception, Mobilenet, Darknet, memory use, computational complexity, SSD Mobilenet, embedded devices, and accuracy (mAP).

Abstract

Detecting traffic signs is important for many purposes, like ensuring safe driving and identifying illegal driving behavior. This study assesses specialized tools (e.g., Resnet, Inception, Mobile-net, and Dark net) and sophisticated detection models. It improves the trained prior model using the Microsoft Coco dataset for German car sign detection by using adaptive learning. evaluation of image size, computational complexity, speed, memory utilization, and accuracy. The results show that R-Fcn Resnet 101 balances precision and quick performance While Ssd Mobile-net is the fastest and best memory, Yolo Version 2 excels in performance & precision making it perfect for embedded devices, mobile devices, and gadgets.

Introduction

Imu sensors are widely used to collect environmental data, but real-time detection of traffic signs is necessary as pre-mapped navigation systems are often outdated. Combining LiDAR and RGB cameras can improve detection accuracy but demands significant computational resources due to the complexity of 3dimensional cloud handling and sensor calculation. Recently, advances in object detection have emerged in the form of Cnn -based models. These models have proved promising for use cases that are either highly accuracy-oriented or very efficiency-oriented, for instance, fast and accurate detection for autonomous vehicles, or light-weight models that fit into a mobile system. Datasets such as Image-net, Pascal voc and Microsoft Coco are used as benchmarks to evaluate detection models. Traditionally, these detection models have been focused on objects such as cars and people, but lately, they have been extended to include traffic sign detection. This work evaluates eight Cnn models trained in advance on Coco and fine-tuned on the German Traffic Sign Data (GTSD) using transfer learning, which reuses pre-trained weights to adapt efficiently to specialized tasks. Analyze mean avg precision (m AP), inference lap, memory usage, floating-point operations, and image size of integrated architectures Faster R-Cnn, R-Fcn, Ssd, and Yolo version 2 along with feature extractors like Res-Net, Inception, Mobile-Net and Dark net. Findings reveal SR-Fcn deals with performance and precision very well. Ssd performs poorly for detection of small sign. Mobile-net is suited to devices having a resource-constrained device such as mobile and embedded systems.

Related Work

This analysis discusses two views on cutting-edge advancements in deep learning for traffic sign, object detection: traditional traffic sign detection operations and recent innovations in Cnn.

Traffic sign Recognition

Traditionally, the technique of Traffic Sign Detection relied on a sliding-window approach combined with techniques for feature extraction such as HOG with SVM classifiers or LBP with Ada-boost. Though these techniques achieved high accuracy, they were highly computationally intensive and could not be applied for real-time detection. The later approaches that emerged integrated Cnn with multi-scale sliding windows, dilated convolutions, and fully convolutional networks, examples of which are EdgeBox-guided R-Cnn. This approach is high-accuracy but low in terms of computational efficiency.

Cnn for Object Recognition

Introduction of Cnn revolutionized object detection. Early models like R-Cnn employed multi-stage pipelines, extracting object proposals via Selective Search and computing feature maps for each proposal, which was computationally intensive. Improvements that are shared computations across the image but introduced inefficiencies in parameter updates. Fast R-Cnn tackled earlier challenges by introducing Roi Pooling and a single-stage training method, which reduced memory usage and increased speed. It replaced Selective

Citation: Prasad TSL, Yuvaraj P, Bala Shiva B, Gnaneshwar T. Machine Learning Techniques for Crop, Fertilizer Prediction and Disease Detection. GJEIIR. 2025;5(4):086.

Paper	Evaluation (%)				Inference time (FPS)	Hardware	
	Metric	P	D	M		CPU	GPU
Liang et al.(2013) [17]	AUC	100	98.85	92	1-2.5	Intel 4-core 3.7GHz	-
Mathias et al.(2013) [16]	AUC	100	100	96.98	2.5	Intel Core i7 870 3.6GHz	NVIDIA GTX 470
Wang et al.(2013) [17]	AUC	100	99.91	100	0.85	Intel Core i3 3.3GHz	-
Zang et al.(2016) [18]	AUC	99.45	98.33	96.5	*	Intel Core 2 Duo 2.2 GHz	-
Aghdam et al.(2016) [23]	AP	99.89			37.72	-	NVIDIA GTX 980

Table 1: *Test results, processing time, and hardware used in different traffic sign detection systems tested on the GTSDB.*

Figure 1. *Sample images from the dataset.*

Search with a Region Proposal Network, Rpn. This shares the feature maps used in the detection network and is therefore more efficient in generating region proposals directly from feature maps. The computation was further streamlined by using position-sensitive score maps in models such as R-Fcn, while models such as Ssd and Yolo combined all tasks of detection into a single fully convolutional network.

Methodology & experimentation

This paper proposes fine-tuning several pre-trained CNN-based models for the specific task of discovering business signs in images with the GTSDB data set. All four architectures included, Faster R-Cnn, R-Fcn, Ssd, and YOLO ver 2, came with six variations of convolutional feature extractors: Res Net ver 1 50, Res Net ver 1 101, Inception ver 2, Inception Res Net ver 2, Mobile Net ver 1, and Dark-net 19. These feature extractors are highly known in the domain of image analysis. The input images go through these extractors to acquire high-level features for traffic sign detection. Experiments relied on privately available pre-trained models due to time and computational constraints, which were initially trained on the Microsoft Coco dataset and fine-tuned for traffic sign detection using transfer learning. The model is trained to classify traffic signs as either mandatory, prohibitory, or warning signs using their shapes and colors. The selection of meta-architectures and feature extractors is carried out by choosing pre-trained models exsist in the TensorFlow Object Detection API and in Yolo repositories, and each feature extractor is assigned to its corresponding meta-architecture because this kind of testing and training time is rather long.

	Faster R-CNN	R-FCN	SSD	YOLO V2
Resnet V1 50	✓			
Resnet V1 101	✓	✓		
Inception V2	✓		✓	
Inception Resnet V2	✓			
Mobilenet V1			✓	
Darknet-19				✓

Table 2: Comparing feature extractors and architectures. Different combinations of Cnn designs and feature extractors are compared.

Datasets

For instance, the United States, Belgium, and China have developed datasets of business signs. However, this paper discusses the German Traffic Sign Detection Benchmark, which has been extensively used in business sign discovery research. The GTSDB data-set is prized due to realistic business scenes captured along roadways, pastoral roads, and civic thoroughfares under varying conditions that encompass different rainfalls, lighting, and occlusions. The business signs are categorized into three important classes: prohibitory (59.5% in training, 161 in testing), obligatory (17.1% in training, 49 in testing), and peril (23.4% in training, 63 in testing). Another aspect that makes the dataset's popular is its public challenge, where scientists compare discovery models on a leaderboard showcasing state-of-the-art methods; however, processing times are not ranked. This standard has effectively reflected real-world business scenarios, making it a dependable tool for assessing business sign discovery models.

Object Detection Frameworks

It shows the key features of each architecture:

Faster R-CNN

Faster R-Cnn operates by sliding a small network over the point chart generated by the last convolutional layer, producing multiple region proposals at each position. These proposals are defined relative to anchor boxes, which are reference boxes of various aspect ratios and scales. To reduce overlap among the proposals, a Non-Maximum Suppression (NMS) algorithm is applied, which eliminates redundant proposals based on their confidence scores. The top 300 proposals, determined by their objectnes scores, are passed to the detection network, which refines the bounding boxes and classifies the objects within them. During testing, all input images were resized so that their shortest edge measured 600 pixels.

R-Fcn

It is built on the Faster R-Cnn module but only uses fully convolutional networks that do not require repeated computations on every region. To address the challenge between translation invariance in image classification (an object's position doesn't affect its recognition) and translation sensitivity in object recognition, R-Fcn introduces location-sensitive score maps. These maps provide accurate localization while maintaining computational efficiency. Similar to Faster R-Cnn, R-Fcn follows a two-stage method that involves region proposal and categorization, with potential regions generated by a fully convolutional Region Proposal Network. The training configuration and hyperparameter settings, including optimization settings and region proposals, are nearly identical to those used in Faster R-Cnn.

Ssd

Unlike others, Ssd is the feed-forward convolutional neural network that doesn't instrument any kind of bounding box proposal or rescoped feature. The Ssd deploys a single-shot architecture for the detection of objects with threading in the detect objects in each single frame of an image. It is achieved by achieving any size or shape for the virtual boxes or anchors. As it is, split into several scales and aspect ratios, each corresponding to bounding box outputs off of a single grid cell. Making use of the scores showing an object's presence and which class it belongs to, given the known default boxes, the network predicts this information to be able to detect objects of different sizes. To achieve different resolutions to predict in the feature map for the Ssd, the network created a number of letters on top of previous layers producing much smaller feature maps that

have made it respond to various-sized objects to be detected. RMSprop had been the optimizers used where there is 5% decay each for 800,000 iterations of a decay. To the input image resized to fit consistent measures was taken with 300 by 300 pixels. Compared to Faster regions convolving neural network or R-Fcn, Ssd is relatively simple, faster, and has much fewer computational requirements.

Yolo version-2

You Only Look Once Version 2 improves upon the ideas and concepts brought out in Faster R-Cnn. It develops those concepts in such a way that it avoids any arbitrary choice prevalent in other object detection technologies. Instead, anchor boxes are optimized during training with K-- means clustering, which is described as follows. It applies a customized distance metric for the selection of anchor boxes that promote the learning of the network. To further refine predictions, Yolo Version 2 calculates bounding box dimensions as offsets from the centroids of these anchor boxes, ensuring they remain within valid ranges of 0 to 1 through a logistic activation function.

Distance (Box, Centroid) = 1 - Intersection over Union (Box, Centroid)

Layer	Type	# Maps & neurons	Kernel size/Stride
0	Input	3 m. of 608x608 n.	
1	Conv	32 m. of 608x608 n.	3x3/1
2	Max-Pool	32 m. of 304x304 n.	2x2/2
3	Conv	64 m. of 304x304 n.	3x3/1
4	Max-Pool	64 m. of 152x152 n.	2x2/2
5	Conv	128 m. of 152x152 n.	3x3/1
6	Conv	64 m. of 152x152 n.	1x1/1
7	Conv	128 m. of 152x152 n.	3x3/1
8	Max-Pool	128 m. of 76x76 n.	2x2/2
9	Conv	256 m. of 76x76 n.	3x3/1
10	Conv	128 m. of 76x76 n.	1x1/1
11	Conv	256 m. of 76x76 n.	3x3/1
12	Max-Pool	256 m. of 38x38 n.	2x2/2
13	Conv	512 m. of 38x38 n.	3x3/1
14	Conv	256 m. of 38x38 n.	1x1/1
15	Conv	512 m. of 38x38 n.	3x3/1
16	Conv	256 m. of 38x38 n.	1x1/1
17	Conv	512 m. of 38x38 n.	3x3/1
18	Max-Pool	512 m. of 19x19 n.	2x2/2
19	Conv	1024 m. of 19x19 n.	3x3/1
20	Conv	512 m. of 19x19 n.	1x1/1
21	Conv	1024 m. of 19x19 n.	3x3/1
22	Conv	512 m. of 19x19 n.	1x1/1
23	Conv	1024 m. of 19x19 n.	3x3/1
24	Conv	1024 m. of 19x19 n.	3x3/1
25	Conv	1024 m. of 19x19 n.	3x3/1
26	Route(17)	512 m. of 38x38 n.	
27	Reorg	2048 m. of 19x19 n.	-/2
28	Concat(25,27)	3072 m. of 19x19 n.	
29	Conv	1024 m. of 19x19 n.	3x3/1
30	Conv	40 m. of 19x19 n.	1x1/1

Table 3: *Yolo Version 2 n/w design*

Result

It depicts the performance results from the experiments conducted on traffic sign detection that were discussed above. The test covers several parameters like precision, the number of total parameters, floating-point operations, memory usage, and time taken to process. Along with those, there are steps like NMS. The results have been obtained with a batch size of one and averaging 300 images from the German test set. Tensor Flow is used in this experiment for calculating the parameters, Flops, and other relevant metrics. Though such results are consistent among all the experiments that are conducted, it does not directly correlate with the previous work due to hardware, software, drivers, frameworks, or batch sizes, among others. Aspects such as memory usage, the number of parameters, and Flops do not depend on the platform, hence a suitable comparison basis.

Performance assessment metrics

Average Precision for each class is obtained through the numerical integration of area under the precision-recall curve. Overall performance termed as mean Average Precision is obtained from average of all the APs obtained across the classes. To determine whether the predicted bounding box, BpB_p, is actually a true positive or false positive compared to the ground truth bounding box, BgtB_{gt}, Intersection over Union is used. Some of the variables that make up this process include recall, rr, precision at a certain recall, p(r)p(r), the ground truth bounding box, BgtB_{gt}, and the predicted bounding box, BpB_p.

$$p(r) = \max_{r':r' \geq r} p(r') \quad (2)$$

$$AP = \sum_{k=1}^{N} p(k)\Delta r(k) \quad (3)$$

$$IoU = \frac{area(B_{gt} \cap B_p)}{area(B_{gt} \cup B_p)} = \frac{area(B_{gt} \cap B_p)}{area(B_{gt}) + area(B_p) - area(B_{gt} \cap B_p)} \quad (4)$$

Model	Class	Avg. IoU	Precision	Recall	AP
Faster R-CNN Resnet 50	Prohibitory	82.52	91.38	98.75	98.62
	Mandatory	81.21	70.00	85.71	85.15
	Danger	85.07	79.45	92.06	90.78
Faster R-CNN Resnet 101	Prohibitory	87.29	90.29	98.14	98.13
	Mandatory	85.58	67.65	93.88	93.46
	Danger	87.05	85.51	93.65	93.64
Faster R-CNN Inception V2	Prohibitory	82.73	81.22	99.38	99.36
	Mandatory	79.66	62.50	81.63	80.47
	Danger	85.62	81.69	92.06	92.03
Faster R-CNN Inception Resnet V2	Prohibitory	91.37	96.99	100	100
	Mandatory	89.16	79.31	93.88	93.66
	Danger	90.11	92.19	93.65	93.65
R-FCN Resnet 101	Prohibitory	87.93	84.66	99.38	99.37
	Mandatory	85.37	76.67	93.88	92.58
	Danger	86.95	86.76	93.65	93.52
SSD Inception V2	Prohibitory	81.76	96.95	78.88	78.77
	Mandatory	80.85	90.00	55.10	54.46
	Danger	85.76	93.18	65.08	65.05
SSD Mobilenet	Prohibitory	80.49	92.50	68.94	67.03
	Mandatory	78.51	89.65	53.06	52.01
	Danger	81.11	79.63	68.25	65.85
YOLO V2	Prohibitory	73.96	92.31	89.44	88.73
	Mandatory	74.66	79.07	69.39	65.70
	Danger	75.82	94.55	82.54	82.06

Table 4: Traffic Sign Detection Accuracy (%)

Model	mAP	FPS	Memory (MB)	GigaFLOPS	Parameters (10^6)
Faster R-CNN Inception Resnet V2	95.77	2.26	18250.45	1837.54	59.41
R-FCN Resnet 101	95.15	11.70	3509.75	269.90	64.59
Faster R-CNN Resnet 101	95.08	8.11	6134.71	625.78	62.38
Faster R-CNN Resnet 50	91.52	9.61	5256.45	533.58	43.34
Faster R-CNN Inception V2	90.62	17.08	2175.21	120.62	12.89
YOLO V2	78.83	46.55	1318.11	62.78	50.59
SSD Inception V2	66.10	42.12	284.51	7.59	13.47
SSD Mobilenet	61.64	66.03	94.70	2.30	5.57

Table 5: Model Ranking by mAP

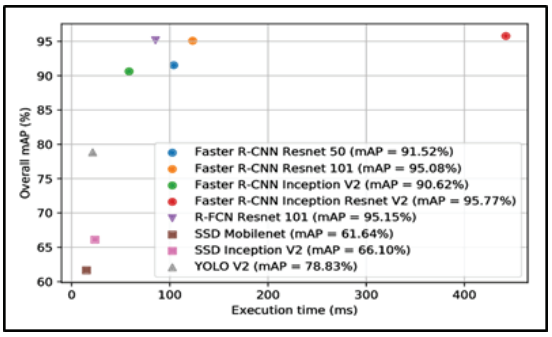

Figure 2. Accuracy / Speed

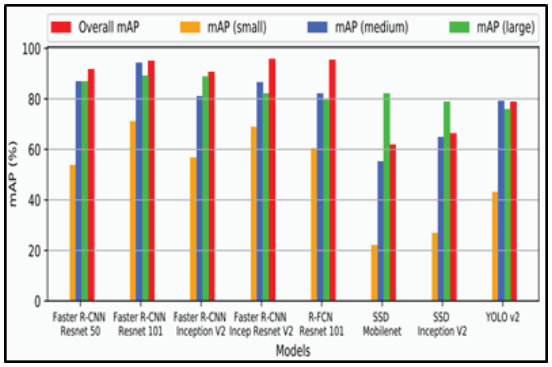

Figure 3. Detector Accuracy by Sign Size

Analysis

The size of traffic signs is the factor that affects the performance of models like Yolo Ver 2 and Ssd; they perform worse on smaller sizes, although these models are shown to be performing better or similarly on larger sizes. This trend was also observed in previous research, in which Yolo Ver 2 could not detect small objects, like plants and bottles. Figure 4 Illustrates the residual network FLOP count in terms of execution time. The denser residual blocks within Faster R-Cnn and R Fcn incur higher Flops and runtimes s while Ssd Mobile-net has the least Flops but remains the fastest acting. However, FLOP counts aren't always equal to execution time; there is a little hardware optimization, and even in the case of YoloVer 2, this runs faster than Ssd Inception Ver 2, whereas the FLOP count seems to be so high. Figure 5 shows that Res-net 101-based models, despite having more number of parameters don't necessarily yield higher execution times compared to that of YOLO Ver 2 with relatively lower parameters.

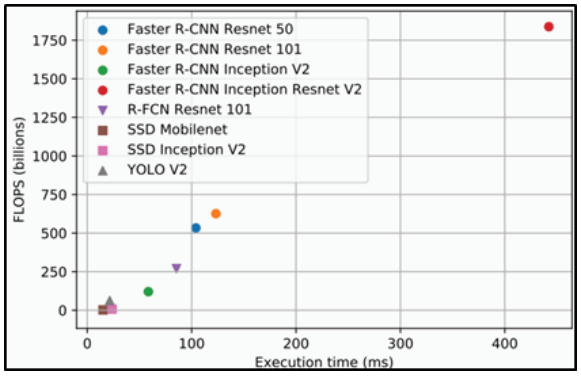

Figure 4. FLOPS / Speed

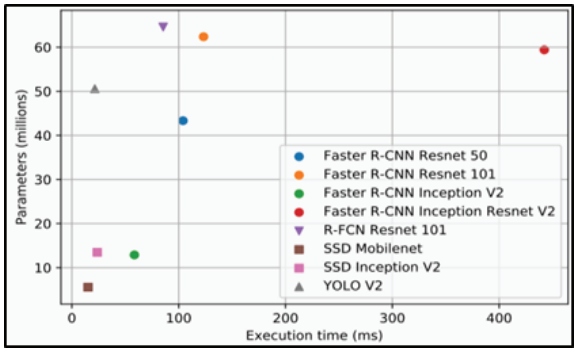

Figure 5. Parameters / Speed

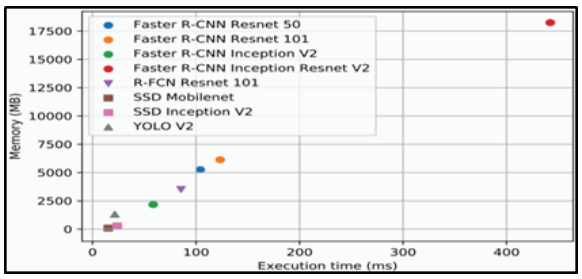

Figure 6. Memory / Speed

Real-World Traffic Sign Detection Output

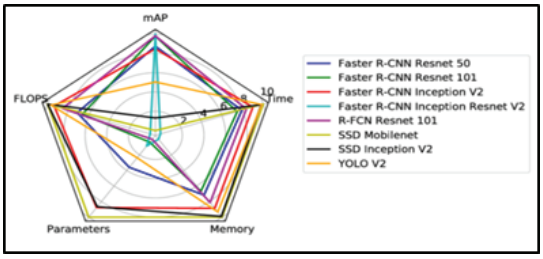

Figure 7. Traffic Sign Detection Analysis

Figure 8. 8 Models with Small, Medium, and Large challengingTraffic Signs detetction

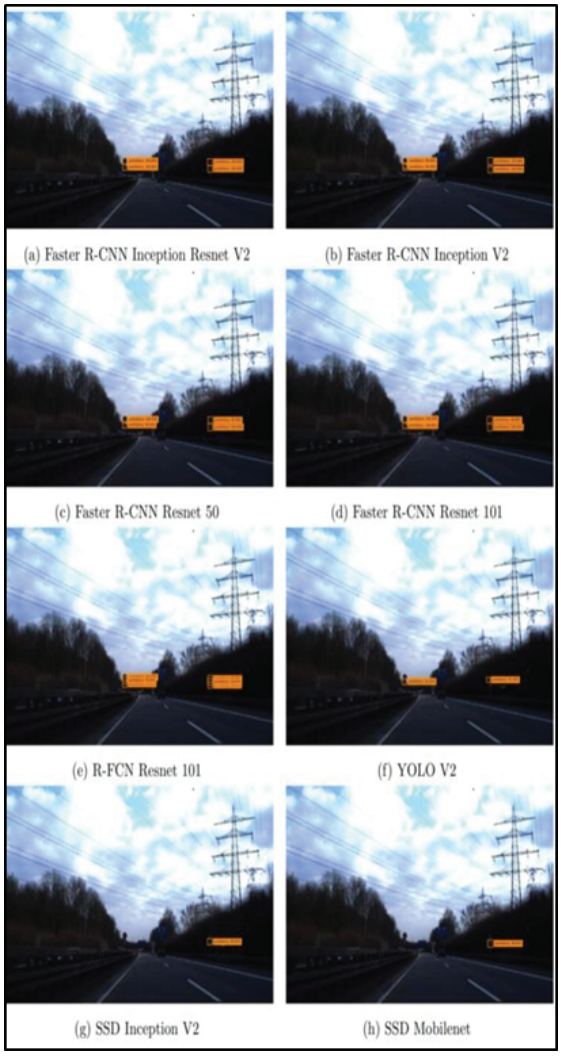

Figure 9. Detection Examples from 8 Models in a Challenging Road Scene

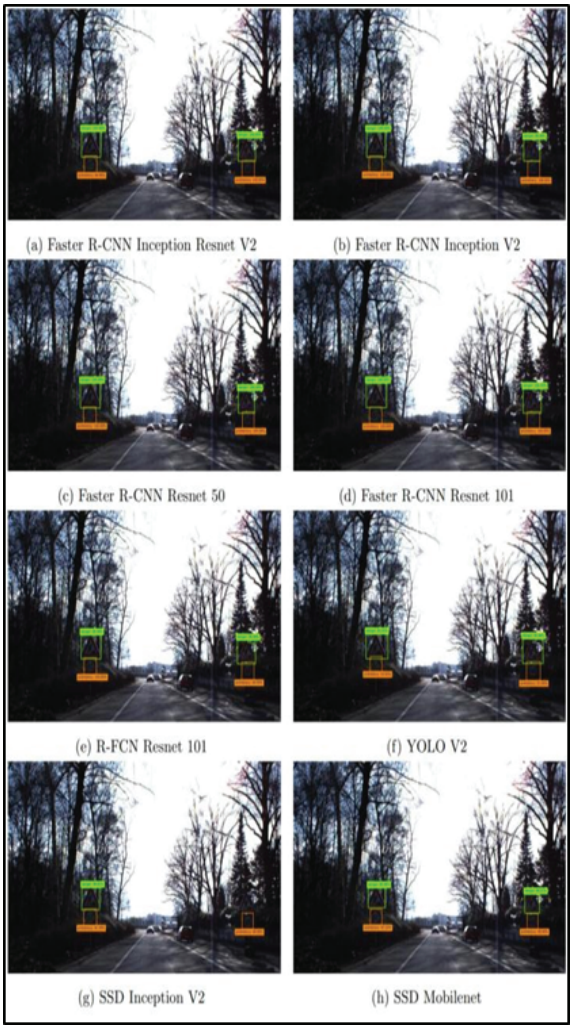

Figure 10. Eight models detect traffic signs in a road scene with signs at both sides. Only group G has an undetected sign

Conclusion

In this research, eight deep neural network-based traffic sign detectors are experimentally compared based on learnable parameters, accuracy, speed, memory utilisation, and floating point operations (FLOPs). To detect traffic sign super classes (required, prohibitory, and hazardous). R-Fcn Res net 101 had the best trade-off between speed (85.45 ms per image) and accuracy (95.15%), whereas the Faster R-Cnn Inceptions Res net Version 2 had the best m AP of 95.77%. The fastest were Ssd Mobile-net and Yolo Ver 2, with Ssd Mobile-net requiring the least amount of memory (15.14 ms per image) and Yolo Ver 2 offering competitive accuracy (78.83%). Yolo Ver 2 and Ssd had trouble with minor traffic signs despite their speed. This is essential for systems operating in real time. With mAPs above 75%, all models did well on huge traffic signs. With over 30 frames per second, Yolo Ver 2 and Ssd are appropriate for real-time applications. Transfer learning produced results that were comparable to the most advanced techniques for detecting traffic signs.

Future Work

Other more advanced neural network architectures that can be proven to have better generalization in object detection should be pursued and adapted to this problem, such as DenseNet. These detectors must be tested on real-world operating conditions through the use of contemporary embedded platforms like the NVIDIA Jetson TX2 and NVIDIA Drive PX7. While the paper had static images of road traffic, the advancements made on detectors would have been noticed if the tracking mechanisms were involved and sequences of frames were applied in video data.

References

1. "DFR-TSD: A Deep Learning Based Framework for Robust Traffic Sign Detection Under Challenging Weather Conditions,"S. Ahmed, U. Kamal and M. K. Hasan in IEEE vol. 23, no. 6, pp. 5150-5162, June 2022. https://ieeexplore.ieee.org/document/9345465
2. "Challenging Environments for Traffic Sign Detection: Reliability Assessment under Inclement Conditions", D. Temel, T. Alshawi, M-H. Chen, and G. AlRegib, in arXiv:1902.06857,2019. https://ar5iv.labs.arxiv.org/html/1902.06857
3. "Multiscale Traffic Sign Detection Method in Complex Environment Based on YOLO v4", Wang, Yongjie, Bai, Miaoyuan, Mingzhi, Zhao, Fengfeng, Guo, Jifeng, 5297605, 2022. https://doi.org/10.1155/2022/5297605
4. "Traffic Sign Detection Under Challenging Conditions: A Deeper

Look into Performance Variations and Spectral Characteristics", in IEEE, D. Temel, M. -H. Chen and G. AlRegib, vol. 21, no. 9, pp. 3663-3673, Sept. 2020.

5. https://ieeexplore.ieee.org/document/8793235
6. Kaggle Source for datasets and discussions related to German Traffic Sign Detection Benchmark dataset (GTSDB).
7. https://sid.erda.dk/public/archives/ff17dc924eba88d5d01a807357d6614c/published-archive.html R. Bhallamudi et al., "Deep Learning Model for Resolution Enhancement of Biomedical Images for Biometrics," in Generative Artificial Intelligence for Biomedical and Smart Health Informatics, Wiley Online Library, pp. 321–341, 2025.
8. R. Bhallamudi et al., "Artificial Intelligence Probabilities Scheme for Disease Prevention Data Set Construction in Intelligent Smart Healthcare Scenario," SLAS Technology, vol. 29, pp. 2472–6303, 2024, Elsevier.
9. R. Bhallamudi, "Improved Selection Method for Evolutionary Artificial Neural Network Design," Pakistan Heart Journal, vol. 56, pp. 985–992, 2023.
10. R. Bhallamudi et al., "Time and Statistical Complexity of Proposed Evolutionary Algorithm in Artificial Neural Networks," Pakistan Heart Journal, vol. 56, pp. 1014–1019, 2023.
11. R. Krishna et al., "Smart Governance in Public Agencies Using Big Data," The International Journal of Analytical and Experimental Modal Analysis (IJAEMA), vol. 7, pp. 1082–1095, 2020.
12. N. M. Krishna, "Object Detection and Tracking Using YOLO," in 3rd International Conference on Inventive Research in Computing Applications (ICIRCA-2021), IEEE, Sept. 2021, ISBN: 978-0-7381-4627-0.
13. N. M. Krishna, "Deep Learning Convolutional Neural Network (CNN) with Gaussian Mixture Model for Predicting Pancreatic Cancer," Springer US, vol. 1380-7501, pp. 1–15, Feb. 2019.
14. N. M. Krishna, "Emotion Recognition Using Skew Gaussian Mixture Model for Brain–Computer Interaction," in SCDA-2018, Textbook Chapter, ISBN: 978-981-13-0514, pp. 297–305, Springer, 2018.
15. N. M. Krishna, "A Novel Approach for Effective Emotion Recognition Using Double Truncated Gaussian Mixture Model and EEG," I.J. Intelligent Systems and Applications, vol. 6, pp. 33–42, 2017.
16. N. M. Krishna, "Object Detection and Tracking Using YOLO," in 3rd International Conference on Inventive Research in Computing Applications (ICIRCA-2021), IEEE, Sept. 2021, ISBN: 978-0-7381-4627-0.
17. T. S. L. Prasad, K. B. Manikandan, and J. Vinoj, "Shielding NLP Systems: An In-depth Survey on Advanced AI Techniques for Adversarial Attack Detection in Cyber Security," in 2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS), IEEE, 2024.
18. S. Sowjanya et al., "Bioacoustics Signal Authentication for E-Medical Records Using Blockchain," in 2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS), vol. 1, IEEE, 2024.
19. N. V. N. Sowjanya, G. Swetha, and T. S. L. Prasad, "AI Based Improved Vehicle Detection and Classification in Patterns Using Deep Learning," in Disruptive Technologies in Computing and Communication Systems: Proceedings of the 1st International Conference on Disruptive Technologies in Computing and Communication Systems, CRC Press, 2024.
20. C. V. P. Krishna and T. S. L. Prasad, "Weapon Detection Using Deep Learning," Journal of Optoelectronics Laser, vol. 41, no. 7, pp. 557–567, 2022.
21. T. S. L. Prasad et al., "Deep Learning Based Crowd Counting Using Image and Video," 2024.
22. https://www.kaggle.com/datasets/safabouguezzi/german-traffic-sign-detection-benchmark-gtsdb