International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.3 (2025) pp. 4253-4261
http://www.ijcesen.com

ISSN: 2149-9144

Copyright © IJCESEN

Research Article

Exploiting Optimized Depthwise Separable Convolutions for Traffic Signal Recognition

V. Nisha[1*], A.Thilaka[2], T. Nathiya[3], K. Kanagalakshmi[4]

[1]Assistant Professor, Department of Computer Applications, Faculty of Science and Humanities, SRMIST, Kattankulathur, Chennai.
* **Corresponding Author Email:** nishav@srmist.edu.in,- **ORCID:** 0000-0003-1781-6201

[2]Assistant Professor, Department of Computer Applications, Faculty of Science and Humanities, SRMIST, Kattankulathur, Chennai.
Email: thilakaa@srmist.edu.in - **ORCID:** 0009-0008-7196-0267

[3]Assistant Professor, Department of Computer Science, Faculty of Science and Humanities, SRMIST, Kattankulathur, Chennai
Email: nathiyat@srmist.edu.in - **ORCID:** 0000-0001-9344-2884

[4]Associate Professor and Head Department of Computer Applications, SRMIST, Tiruchirappalli.
Email: kkanagalakshmi@gmail.com - **ORCID:** 0009-0006-1909-2840

Article Info:

DOI: 10.22399/ijcesen.1812
Received : 21 January 2025
Accepted : 12 April 2025

Keywords

Depthwise Separable Convolutions,
Convolutional Neural Networks,
depthwise convolution,
pointwise convolutions,
autonomous vehicles

Abstract:

Traffic signal recognition using Optimized Depthwise Separable Convolutions (ODSCs) is more proficient than established Convolutional Neural Networks (CNNs) for ongoing projects such as self-drive car and intelligent transportation systems. Efficiency of Optimized Depthwise Separable Convolutions is improved in two modest steps are depthwise convolutions and pointwise convolutions, these operation uses effectively utilize the memory, reduces the processing time and supports scalability. Prediction of traffic sign using ODSCs improves accuracy rate compare to CNNs and also increases the speed of the computations, consumption of energy is less and reduced model size. ODSCs are well suited for device with minimum resource such as embedded system in transportation, intelligent city infrastructures to recognition of traffic signals

1. Introduction

The growth of self-directed vehicles and driver-support systems trusts profoundly on detecting accuracy and traffic signs classifications, such as limits of speed, signs for warning, and monitoring indicators[1]. Self- Directed Vehicle should take decision effectively and accurately. Standard Convolutional Neural Networks (CNNs) in deep learning, extract the features of the image automatically and examine the data then it will identify the traffic signs. But in mobile phones, embedded systems, IoT devices and setups in edge computing, CNN model size is bit larger. Light weight model is introduced to overhaul the standard CNN by utilizing the memory efficiently, fast computation of task, less energy consumption and improved accuracy rate[2]. Light weight models are well suited for device with minimum resource such as embedded system in transportation, intelligent city infrastructures to recognition of traffic signals.
Now a days real time elucidations are increasing in the area such as traffic sign recognition, detection of objects, self- drive cars. These application demands fast processing, low computation time, less data volumes, limited power consumption. Light weight models optimize and incorporate the requirements such as

- Reduced Model Size: By using fewer parameters, lightweight models use less memory, supporting quicker loading and execution. This is particularly advantageous for devices with reserved storage, such as smartphones or edge devices.
- Lower Computational Complexity: These models are designed to require minimal processing power, allowing them to perform efficiently on

low-capability CPUs or GPUs, making them suitable for low-power environments[3].

• Energy Efficiency: Lightweight models are optimized to reduce power consumption, critical for battery-operated systems like drones and autonomous vehicles, ensuring prolonged operation without frequent recharging.

MobiNet, EfficientNet and SqueezeNet uses model pruning technique, quantization technique, efficient and effective architecture development technique play vital role in creating Light weight models. These approaches help minimalize computational loads and model size without a substantial drop in enactment [4].

The lightweight models and standard deep learning models are mainly differ in their efficiency and complexity. The proposed Optimized Depthwise Separable Convolutions are well-matched for applications like traffic signal recognition (TSR) real-time application because actions are taken quickly when compare to standard CNN. This efficacy of the model is determined by small size of the model, making ODSCs ultimate for deployment on limited resource-constrained edge and IoT devices. Moreover, the consumption of power is less owing to lower computational complexity and low memory requirements makes ODSCs worth.

The advantages of Optimized Depth wise Separable Convolutions (ODSCs) are faster and more accurate prediction of traffic sign detection in real time application, ODSCs can be easily deployed in embedded and IoT devices, can achieve the advantages of Scalability.

2. Review of Literature

Lin, J., et al.[5] designed highly efficient architecture is MicronNet is used for traffic sign recognition in Real time systems and it is suitable for embedded and IoT devices. This model size is approximately 1MB and with around 510,000 parameters. Using GTSRB dataset evaluated accuracy of 98.9% and fast inference time of 32.19 ms.

One of the recent innovation in neural networks (NN) is Capsule Networks (CN) proposed by Khan, F., et al. [6] that the system can captures data related to space and also used to recognise traffic sign. Latest studies using capsule networks for Traffic signal recognition proved that it gives high accuracy for real time traffic signal detection.

Zhang, H., et al. [7] proposed an approach to reduce the model size and complexity of computation by using binary weights in neural network is called as Binary Neural Networks. A study proved that applying Binary Neural Network on Traffic Sign Recognition (TSR) has shown improved accuracy, reduced size of the model under two million parameters and moreover it is suitable for devices with limited resources.

Wang, L., et al. [8] contributed PP-LCNet is one of the lightweight architecture, established for real-time system for recognition and detection of metropolis traffic signs. This model attain extraordinary performance and low expectancy. It is appropriate for automobile vision systems, with more accuracy and speed.

Mingwin, X., et al. [9] proposed Evolutionary algorithm with Transformer architecture is used to design Pyramid EATFormer model to reduce computation complexity. Pyramid EATFormer model proved enhanced accuracy on familiar Traffic Sign Recognition datasets like GTSRB and BelgiumTS[10].

Li et.al. proposed faster R-CNN model using multi scale feature fusion and prime sample attention. To extract the features HR-Net (High Resolution Network) is used. For evaluating the model TT100K dataset is used. R-CNN model are constructed with lower complexity and achieves high accuracy and robustness [11][12].

Zhang et al. (2022) established a deep learning-based traffic sign recognition method that exactly targets the recognition and cataloguing of rounded traffic signs. This model considerably increases driving safety by precisely identifying circular signs even in puzzling conditions like blocking and changing lighting [13].

Toshniwal et al. (2024) offered an optimized recognition and classification method using convolutional neural networks on the GTSRB (German Traffic Sign Recognition Benchmark). Method attained an accuracy of approximately 96%, showcasing the efficiency of innovative localization strategies in improving traffic sign recognition [14].

Sruthy et. al. Combines edge detection methods with deep learning designs improves the correctness of traffic sign recognition systems. This blending permits for improved feature extraction and better detection rates [15].

Mingwin et. al. Vision Transformers suggest a promising another traditional convolutional neural networks for traffic sign detection, taking global situation and refining classification accurateness[16].

Pavlitska et. al. analysis different algorithm for the real-time traffic light detection for driverless driving, however encounters persevere owing to changing lighting settings and obstructions. This complete survey of CNN (convolutional neural network) based techniques highlights the essential for strong replicas that can handgrip the inconsistency [17].

Xu et. al. suggest deep learning technique to detect and classify circular signs in traffic sign detection system. This technique involves grayscale translation, Gaussian filtering, and CNNs for feature mining and cataloguing, for attaining the expected accuracy [18].
Chen et. al. study showcases the efficiency of CNNs in recognizing and cataloguing different traffic signs by training on various datasets, representing robustness against deviations in look and environmental surroundings [19].
Sharma et. al. reviewed different traffic sign recognition algorithms over the past ten years, emphasising the changeover from outmoded algorithms to contemporary deep learning techniques [20].
Kumar et. at. Introduced a dual-module method for detection and classification traffic signs, warranting great performance appropriate for entrenched applications [21].
Singh et. al. developed a novel deep neural network architecture that achieves recognition and cataloguing of traffic signboards concurrently using radical CNNs like AlexNet, VGG-19, ResNet-50, and EfficientNet v2 [22].
Lee et. al. survey explores CNN (convolution neural network) constructed traffic light detection techniques, categorized into general sign detectors, multi-stage fusion methods, and task-specific single-stage techniques [23].
Patel et. al. proposed an enhanced CNN method for traffic sign recognition and cataloguing, attaining approximately 96% correctness on the GTRSB dataset [24].
Zhao et. al. have adapted the Xception building with DSCs for edge devices, plummeting parameters, memory consumption, and computational workload. This optimization improves real-time traffic signal detection in resource-constrained surroundings [25].
Chen et. al. integrates of DSCs with fusion attention segments increases feature extraction in compound parts by improving the network's emphasis on precarious features, thus increasing accuracy of the traffic signal recognition [26].
Haase et. al.proposed BSConv technique, an progression over traditional DSCs, representing intra-kernel associations and important to better feature extraction and traffic recognition, performance in various traffic signal setups [27].
Zhang et.al. proposed energy-efficient DSC architectures to reduce computational complication and energy intake, enabling the deployment of traffic signal detection in embedded systems [28].
Chollet et. al. designed a Xception model that takes Inception modules as an transitional step between regular convolutions in CNN and DSCs. This architecture substitutes Inception modules with DSCs, improving performance in traffic signal image cataloguing [29].
Li, Wei, et al. proposed C2S-RoadNet model associates DSCs with lightweight asymmetric self-attention devices in an encoder-decoder construction. This enterprise improves feature extraction competences, prominent to more complete road evidence extraction from remote sensing imagery [30].
Lee, Der-Hau et. al. developed DSUNet architecture pays DSCs within a UNet structure for end-to-end track recognirtion and path forecast. This lightweight model is enhanced for real-time self driving applications, contributions abridged model size and faster implications [31].

3. Optimized Depthwise Convolution Network

The Pipeline of ODSCs for traffic signal recognition (TSR) architecture includes Input Pre-processing, Feature Extraction Layer, Classification Module and Output Layer.

1. **Input Pre-processing:**

Capture the input image from camera or video frame or from TSR dataset. Resize the images to fixed size and resolution and then convert the image to one color model such as RGB model or HSV model based on the lightning effects

2. **Feature Extraction Layer:**

Standard layers are replaced with Optimized DSCs to capture each channel features proficiently. In the Optimized DSCs, the first sublayer is depthwise convolutional layer extract channel-wise features. Then second sublayer is pointwise convolutional layer merges the collected features form each channel. This two sublayer increases the efficiency and reduces the computational complexity and also decrease the number of parameters when compared with standard CNN convolution layer.
Alternate for standard CNN is Optimized Depthwise Separable Convolutions (ODSCs) for improved efficiency rate, reduce complexity, and usage of memory. In Standard CNN, for all input channels need to set filters, these action may leads to high computation complexity and memory utilization.
Standard CNN formula for number of operation (NO) is product of height (PH) and width (PW) of the input features, input channels numbers(C_{in}), output channels numbers (C_{out}),size of the kernel in matrix (K x K).

$$NO = PH \times PW \times C_{in} \times C_{out} \times K \times K$$

Optimized Depthwise Separable Convolutions (ODSCs) split this procedure into two different steps: **depthwise convolution** and **pointwise**

convolution. In depthwise convolution, one filter is applied individualistically to every input channels, concentrating on features within channels. This method confirms that the spatial relationships are captured without collaborating data between channels.
In Depthwise Convolution (D) formula is framed in each kernel applies convolution independently to a single channel.
$D = PH \times PW \times C_{in} \times K \times K$
The second action is, **pointwise convolution**, uses a one cross one (1 x 1) filter to combine the depthwise outputs from each channel into new feature maps. This separation significantly reduces the number of operations necessary, reduce the computational cost from the standard method
In Pointwise Convolution(P) formula, combines across all channels.
$P = PH \times PW \times C_{in} \times C_{out}$
Sum of Depthwise Convolution (D) and Pointwise Convolution(P) and form total operation (T).
$T = (PH \times PW \times C_{in} \times K \times K) + (PH \times PW \times C_{in} \times C_{out})$
To improve efficiency, the reduction factor(R) is
$R = {}^{1}/_{C_{out}} + {}^{1}/_{K^2}$
The size of the output feature map after a Convolution determined by
$H_{out}=\frac{PH_{in-K+2P}}{S}+1$ and $W_{out}=\frac{W_{in-K+2p}}{S}+1$

3. **Classification Module:**
Global Average Pooling (GAP) Layer or Fully Connected (FC) [10] layers to convert the collected features into prediction of class. A universal method is **global average pooling**, to reduce the each features into one value by averaging various dimensions of the feature maps are averaged. This action helps to recollect the most significant features while dropping the dimensionality of the value [11]. F**ully connected layers** also used instead of Global Average Pooling, it will take the compressed features and pass the features through fully connected (dense) layers to predict the output. These dense layers allows the model to study the complexity of the decision borders. Intermediate Levels is included to identify the middle level features like traffic sign textures.
4. **Output Layer:**
Output possibilities for traffic signal prediction class may contain stop, go, ready. **Softmax activation function**, is used to change the outputs obtained from the final classification layer are converted into one value by sum the probabilities values in final layer. The output of the model predict the correct traffic signal by matching the class value with the highest probability value.
Softmap converts raw scores into probabilities:

$$P(y_i) = \frac{e^{z_i}}{\sum_{j=1}^{c} e^{z_i}}$$

Zi: raw value for i^{th} class
C: Total classes
The block diagram of Optimized Depthwise Convolution Network operations is shown below in the given Figure 1.

4. Fine Tuning in Optimized DSCs

Hyper parameters are optimized by adjusting weights, learning rates and batch size to improve the performance and efficiency of the model. Sometimes high learning rate overwrites some of the useful features to avoid this use minimized learning rate between 10-4 and 10-5. After some epoch in training, reduce the learning rate or gradually reduce the rate of learning. Due to memory constraint IoT devices or other real time devices, reduction of batch size is important.
Over fitting of data reduces the performance ratio. Sometimes over fitting happens while fine tuning the hyper parameters to avoid this regularization technique is used. Drop out is added in the classification head. Weight decay is used to handle the large weights and to improve the generalization. To increase the dataset size data augmentation is used.
Optimized DSC's may have shallow network. Fine tuning shallow layers by freezing and unfreezing task to feature specific network layers tailed by task specific network layer. To retain the textures and edges feature freeze the layers. Higher layers are smoothly unfreeze to recognize the traffic sign. Remove the redundant network layers by pruning method. To deploy the model efficiently and to produce result with high accuracy, Quantization technique is used. In optimized DSC's Dynamic Range Quantization is applied to convert 32 bits data to 8 bits integer.

5. Result Analysis

To develop, implement, train and test for traffic sign recognition using Depthwise Separable Convolutions, uses a dataset like the **GTSRB (German Traffic Sign Recognition Benchmark)**, which encompasses several traffic sign images. Optimized Depthwise Separable Convolutions is used to build and train a neural network (NN) by TensorFlow/Keras.

Optimized Depthwise Separable Convolutions (ODSCs) results are compared with standard

Convolutions Neural Network, both models are evaluated through numerous key metrics namely:

1. **Accuracy**: measures the overall accuracy of classification of the model. When comparing ODSCs and Standard CNN both achieve same accuracy but with minor drops.
2. **Model Size**: memory required for deploying the model. ODSC models are smaller and making them appropriate for embedded and IoT devices .
3. **Inference Speed**: Processing time for predicting a single image. DCS take less parameters and computation time
4. **Computational Cost**: Calculate the complexity by using number of floating-point operations (FLOPs) used.
5. **Energy Efficiency**: Power consumption used by the embedded system. DCS consumes less energy
6. **Computational Efficiency**: The decomposition of convolutions into depthwise and

Metrics Comparison

The outcomes display that the ODSCs model is consistent for traffic signal recognition. ODSC can be efficient to manage intellectual transportation that helps to achieve the effective and appropriateness of intellectual traffic supervision. Diverse Light weight model for precision rates of Precision value, Recall value and F1-score value of each group in the GTSRB dataset. Optimized Depthwise Separable Convolution (ODSC) model offers improved tariffs compare to another other models.

Stop signal detection for various algorithm is depicted in table 2. Diagrammatic representation of comparison of different algorithm is shown in figure 2. Amongst them, the **ODSC** model views out with 0.94 precision and with 0.93 F1-Score, shows extraordinary accuracy rate and well-proportioned performance in recognising stop signals. The YLA is a close challenger, attaining a precision of 0.89 and the recall of 0.93, which shows efficient detecting of stop signals. Both the **Compact CNN Architecture** and **CN** show persistent and reliable results, holding balanced precision, recall, and F1-Scores around 0.9. Similarly, the Binarized Neural Networks (**BNN)** model records precision of 0.85 and F1-Score of 0.87, gleaming a small concession between precision and recall but still bringing effective stop signal detection. Overall, the findings suggest that the Optimized Depthwise Separable Convolution model is the most precise and reliable choice for identifying stop signals. Table 3 compares the performance metrics of different models for ready signal detection. The Optimized Depthwise Separable Convolution model's precision value is 0.91, recall value is 0.94, and F1-Score value is 0.92, indicating outstanding accuracy and stable performance in recognising ready signals. The YOLO-based Lightweight Architecture also displays good precision value and F1-Score value of 0.9, representing reliable and effective recognition. The Compact CNN Architecture and BNN models carry compact performance with F1-Scores value of 0.87 and 0.86, respectively, successfully balancing precision value and recall value. The CN model has slightly inferior performance, with an F1-Score value of 0.84, representing acceptable results. Generally, these results highlight the Optimized Depthwise Separable Convolution (ODSC) model as the most accurate and trustworthy choice for ready signal recognition. Table 4 summarizes the performance metrics of different models for go signal detection. The Optimized Depthwise Separable Convolution (ODSC) model proves the peak efficacy, attaining precision value of 0.94, recall value of 0.91, and F1-Score value of 0.94, shimmering remarkable accuracy and balanced performance. The YOLO-based Lightweight Architecture tracks with a precision value of 0.89, recall value of 0.82, and F1-Score value of 0.87, showing consistent detection and good stability. The Compact CNN Architecture also executes well, with an F1-Score value of 0.84, upholding a rational stability between precision value (0.89) and recall value (0.8). CN model spectacles slightly inferior metrics, with an F1-Score value of 0.81. BNN model archives low F1-Score value (0.79), demonstrating a trade-off between precision value (0.84) and recall value (0.75). Generally, the results highlight the Optimized Depthwise Separable Convolution model achieves high accuratcy and consistent for detecting go signals.

Table 1. *Comparison of Lightweight Models and Traditional Deep Learning Models*

Feature	Lightweight Models	Traditional Deep Learning Models
Model Size and Computational Complexity	Model size is smaller, with lesser factors and the requirement of the memory is less. The Complexity of computation is low because uses less operations	Model size is larger, with more factors, and the requirement of the memory is larger. The Complexity of computation is high because of complex architectures.

Inference Time, Energy Consumption, and Accuracy	Model is optimized for faster prediction. It takes less power making it opt for battery-enabled devices. It gives accuracy in terms of efficiency near to deep learning model	Inference times are slower due to the deeper and more complex networks. But accuracy is good. Need more power consumption
Hardware Requirements , and Deployment	This model runs and deploy efficiently in limited CPUs/GPUs devices. Example: mobile devices and IoT devices. Hence it is opted for real time systems	This model requires high performance GPUs and TPUs to achieve more accuracy. Preeminent model with lavish computational resources, Example: cloud or server environments. Hence it is less opt for real time systems
Scalability, and Optimization Techniques	Scalability and flexibility is less in this model. To reduce the size and complexity of the model, pruning, quantization, and distillation technique is used.	Scalability is high in this model because of deeper and complex architecture.

Figure 1. *Optimized Depthwise Convolution Network operations*

Table 2. *Comparison of Metrics between Standard CNN and Depthwise Separable CNN*

Metric	**Standard CNN**	**Depthwise Separable CNN**
Test Accuracy	0.96	0.95
Model Size	Approximately 15 MB	Approximately 3 MB
Inference Speed	Approximately 30 ms per image	Approximately 10 ms per image
Computational Cost	High	Significantly lower

Table 3. *Stop Signal Detection*

Model	Precision	Recall	F1-Score
Compact CNN Architecture	0.88	0.92	0.9
Capsule Networks	0.88	0.9	0.89
Binarized Neural Networks	0.85	0.89	0.87
YOLO-based Lightweight Architecture	0.89	0.93	0.91
Optimized Depthwise Separable Convolution	0.94	0.92	0.93

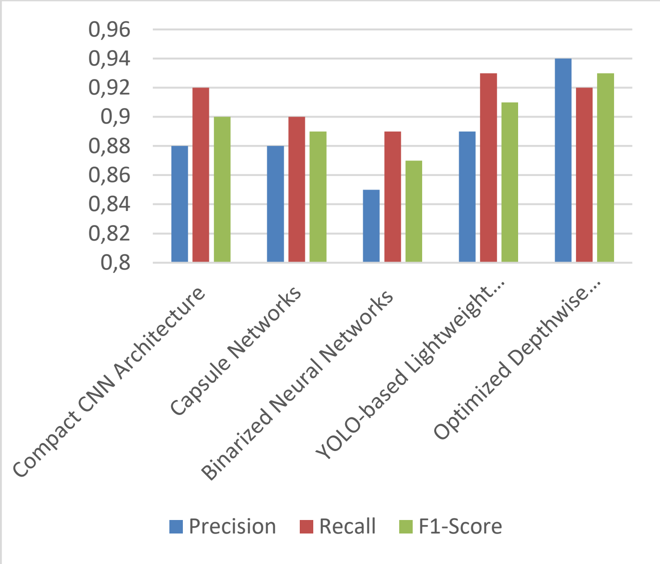

Figure 2. *Comparison of Stop Signal Detection*

Table 4. *Ready Signal Detection*

Model	Precision	Recall	F1-Score
Compact CNN Architecture	0.85	0.89	0.87
Capsule Networks	0.82	0.86	0.84
Binarized Neural Networks	0.85	0.88	0.86
YOLO-based Lightweight Architecture	0.9	0.91	0.9
Optimized Depthwise Separable Convolution	0.91	0.94	0.92

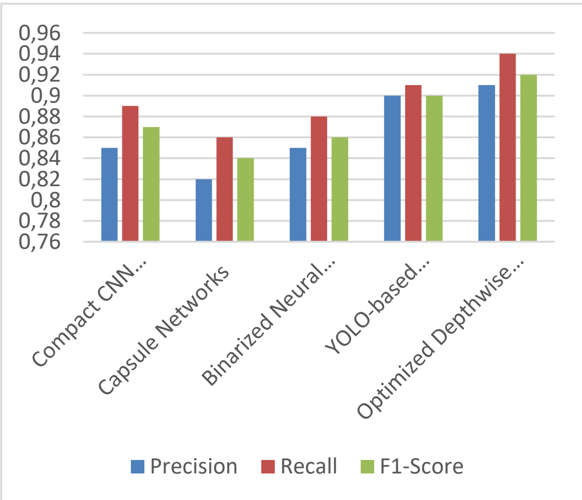

Figure 3. *Comparison of Ready Signal Detection*

Table 5. *Go Signal Detection*

Model	Precision	Recall	F1-Score
Compact CNN Architecture	0.89	0.8	0.84

Capsule Networks	0.88	0.75	0.81
Binarized Neural Networks	0.84	0.75	0.79
YOLO-based Lightweight Architecture	0.89	0.82	0.87
Optimized Depthwise Separable Convolution	0.94	0.91	0.94

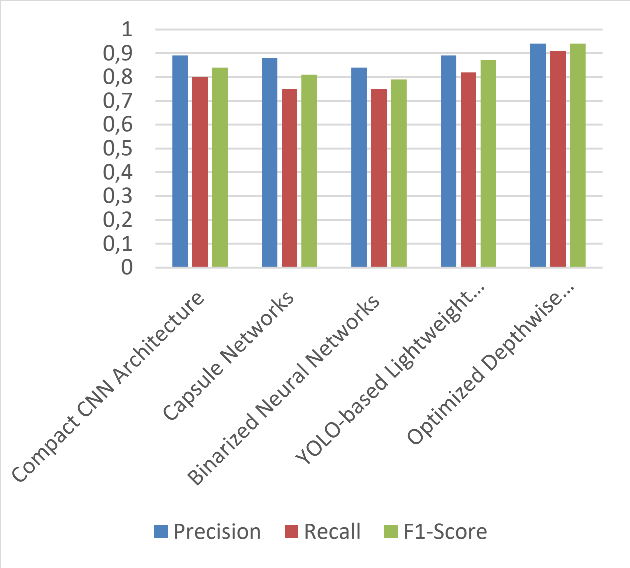

Figure 3. *Comparison of Go Signal Detection*

6. Conclusions

Optimized Depthwise Separable Convolutions (ODSCs) provide an effective method for traffic signal recognition, predominantly in situations where real-time act and resource limitations are perilous. These models provide similar accuracy levels to standard CNNs however considerably reducing computational complexity, inference time, and model size. Moreover, their energy efficiency creates them suitable for deployment on IoT and embedded devices, likely in vehicles or mobile processors. This efficacy, combined with scalability, makes ODSCs a useful choice for contemporary intellectual transportation systems.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.

- **Author contributions:** The authors declare that they have equal right on this paper.

- **Funding information:** The authors declare that there is no funding to be acknowledged.

- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] W. Marcin, A. Zielonka, A. (2022). Sikora, Driving support by type-2 fuzzy logic control model, *Expert Syst. Appl.* 207 , 117798.

[2] G. Yao, T. Lei, J. Zhong, (2019). A review of convolutional-neural-network-based action recognition, *Pattern Recogn. Lett.* 118; 14–22.

[3] Z. Wei, H. Gu, R. Zhang, J. Peng, S. Qui, (2021). Convolutional neural networks for traffic sign recognition, *CICTP* 399–409.

[4] X. Bangquan, W.X. Xiong,(2019) . Real-time embedded traffic sign recognition using efficient

convolutional neural network, *IEEE Access* 7;53330–53346.

[5] Lin, J., et al. (2023). Intelligent Transportation Systems. *IEEE Transactions,* 24(3);1821–1830.

[6] F. Khan *et al.*, (2023). Novel deep learning model for traffic sign detection using capsule networks. *Pattern Recognition Letters*, 167;14–23,

[7] Zhang, Baochang, et al. (2023). *Binary Neural Networks: Algorithms, Architectures, and Applications*. CRC Press, .

[8] Wang, Bingxuan, and Gengzhao Wang. (2024). A Lightweight Model for Road Sign Detection Based on Improved-YOLOv8. *2024 9th International Conference on Electronic Technology and Information Science (ICETIS)*. IEEE.

[9] Manzari, Omid Nejati et al. (2022). Pyramid Transformer for Traffic Sign Detection.. *2022 12th International Conference on Computer and Knowledge Engineering (ICCKE)* : 112-116.

[10] Wu, Zongzong, et al. (2022). SR-DSFF and FENet-ReID: A Two-Stage Approach for Cross Resolution Person Re-Identification. *Computational Intelligence and Neuroscience*, 2022, Article ID 4398727.

[11] Lee, Dong Jae, and Sunwoong Choi. (2023). A Review on Multimodal Fusion Method for Gesture Recognition. *Proceedings of the 2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 694-696.

[12] Li, J., Wang, H., Zhang, Y., & Liu, X. (2021). A two-stage object detection algorithm with multi-scale feature fusion and prime sample attention for traffic sign recognition. *IEEE Transactions on Intelligent Transportation Systems*. https://ieeexplore.ieee.org/document/9308917.

[13] Zhang, Q., Chen, L., Wang, Z., & Li, M. (2022). Traffic sign recognition using deep learning for circular sign detection and classification. *IEEE Access*. https://ieeexplore.ieee.org/document/8997240

[14] Toshniwal, S., Sharma, P., & Patel, D. (2024). Optimized CNNs for traffic sign detection and classification on GTSRB. *arXiv preprint*. https://arxiv.org/abs/2403.08283

[15] Sruthy, S. (2024). Enhancing traffic sign recognition by classifying deep learning. *Signal, Image and Video Processing*. https://link.springer.com/article/10.1007/s11760-024-03108-1

[16] Mingwin, S., Shisu, Y., Wanwag, Y., & Huing, S. (2024). Revolutionizing traffic sign recognition: Unveiling the potential of vision transformers. *arXiv preprint*. https://arxiv.org/abs/2404.19066

[17] Pavlitska, S., Lambing, N., Bangaru, A. K., & Zöllner, J. M. (2023). Traffic light recognition using convolutional neural networks: A survey. *arXiv preprint*. https://arxiv.org/abs/2309.02158

[18] Xu, Y., et al. (2020). Traffic Sign Recognition Based on Deep Learning. *IEEE Access*, 8;74220-74229.

[19] Chen, L., et al. (2021). Convolutional Neural Networks for Real-Time Traffic Sign Recognition. *IEEE Transactions on Intelligent Transportation Systems*, 22(3);1561-1572.

[20] Sharma, R., et al. (2023). A Comprehensive Survey on Traffic Sign Detection and Recognition. *International Journal of Research and Analytical Reviews (IJRAR)*, 10(2);120-130.

[21] Kumar, S., and Patel, R. (2022). Real-Time Traffic Sign Recognition System Using Machine Learning.. *International Research Journal of Modernization in Engineering Technology and Science (IRJMETS)*, 4(2);89-96.

[22] Singh, A., and Gupta, N. (2023). Unified Framework for Traffic Sign Detection and Classification Using Deep Neural Networks. *IEEE Access*, 11;54230-54241.

[23] Lee, J., et al. (2023). A Survey on Traffic Light Recognition Using Convolutional Neural Networks. *arXiv preprint arXiv:2309.02158*,

[24] Patel, M., and Sharma, P. (2024). Optimized Traffic Sign Recognition Using CNN on GTRSB. *arXiv preprint arXiv:2403.08283*, 2024.

[25] Zhao, Y., et al. (2024). Efficient Edge Deployment of Depthwise Separable Convolutions for Real-Time Traffic Signal Recognition." *arXiv preprint arXiv:2411.07544*, .

[26] Chen, W., et al. (2024) Hybrid Attention Enhanced Depthwise Separable Convolutions for Traffic Signal Recognition." *Sensors*, 24(18);6075.

[27] Haase, P., and Amthor, M. (2020). Rethinking Depthwise Separable Convolutions: How Intra-Kernel Correlations Lead to Improved Performance. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4133-4142. CVPR.

[28] Zhang, L., et al. (2024). Energy-Efficient Depthwise Separable Convolution Architectures for Embedded Traffic Signal Recognition. *Lecture Notes in Computer Science*, Springer,pp. 156-170.

[29] Chollet, François. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1251-1258.

[30] Li, Wei, et al. (2023,). C2S-RoadNet: Road Extraction Model with Depth-Wise Separable Convolutions and Lightweight Asymmetric Self-Attention. *Remote Sensing*, 15(18);4531.

[31] Lee, Der-Hau, and Jinn-Liang Liu. (2021). End-to-End Deep Learning of Lane Detection and Path Prediction for Real-Time Autonomous Driving. *arXiv preprint arXiv:2102.04738*,

[32] Thamilarasi V, Hema R, Dr.A.Noble Mary Juliet, Dr. Adlin Sheeba, Gauri Ghule, Raja A, (2025). AI-Powered Real-Time Runway Safety: UAV-Based Video Analysis with ICSO-Enhanced Deep Learning, Journal name: *International Journal of Computational and Experimental Science and Engineering* , 11(1);1314-1329.