

Multi-Task Learning for Traffic Sign Recognition using Multi-Scale Convolutional Neural Networks

Mutaqin Akbar¹, Indah Susilawati¹, Budi Sulistiyo Jati¹, Nur Alamsyah²

¹Department of Informatics, Universitas Mercu Buana Yogyakarta, Indonesia

²Department of Information System, Universitas Informatika dan Bisnis Indonesia, Indonesia

Article Info

Article history:

Received May 22, 2025

Revised Jul 10, 2025

Accepted Aug 15, 2025

Keywords:

Convolutional neural network

Multi-scale

Multi-task

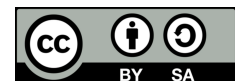
Recognition

Traffic sign

ABSTRACT

Traffic signs are an essential component of road infrastructure. According to the Department of Transportation, Indonesia has over 300 distinct traffic signs, categorized based on their functions and purposes. TSR systems have been widely integrated into various intelligent transportation technologies, such as Driver Assistance Systems (DAS), Advanced Driver Assistance Systems (ADAS), and Autonomous Driving Systems (ADS). The output generated by TSR serves as a critical input for DAS, ADAS, ADS, and other intelligent systems. This article presents a CNN-based classification for traffic sign recognition using multi-task learning (MTL), focusing on traffic signs in Indonesia. The dataset was collected from direct capture with the help of a cellphone camera, indirect capture by utilizing screenshots on a digital map application, and they are captured from several different angles, during the day and at night. The proposed CNN architecture incorporates multi-scale within an MTL framework. The use of a multi-scale approach will hopefully enhance the model's ability to recognize traffic signs in varied and complex environments. And the integration of MTL will enable the model to handle multiple related tasks concurrently, sharing learned features across tasks. During the training stage, the MS-CNN outperformed a standard CNN model by demonstrating lower initial loss, higher starting accuracy, and achieving 100% accuracy by the 8th epoch with a minimal error rate of just 0.003. In the testing stage, the model achieved exceptional results, as shown by the confusion matrix, it successfully classified all traffic sign types (10 classes) and accurately categorized each sign into one of two categories—warning or prohibition. All performance metrics, including precision, recall, and F1-score, reached 100% for both output tasks, confirming the robustness and reliability of the model.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Mutaqin Akbar,

Department of Informatics,

Universitas Mercu Buana Yogyakarta,

Jl. Jembatan Merah No.84C, Depok, Sleman, Daerah Istimewa Yogyakarta 55283, Indonesia.

Email: mutaqin@mercubuana-yogya.ac.id

1. INTRODUCTION

The Indonesian government, through the issuance of Regulation Number 13 of 2014 by the Minister of Transportation, has established comprehensive guidelines for traffic signs across the country. Traffic signs are an essential component of road infrastructure, consisting of symbols, letters, numbers, or a combination thereof, designed to convey warnings, prohibitions, commands, or instructions to road users. According to the Department of Transportation, Indonesia has over

300 distinct traffic signs, categorized based on their functions and purposes, including warning signs, prohibition signs, command signs, recommendation signs, and guide signs. One of the cutting-edge technologies employed for the identification and interpretation of these traffic signs is Traffic Sign Recognition (TSR). By leveraging advanced image processing techniques, TSR enables the automatic detection and classification of traffic signs. TSR systems have been widely integrated into various intelligent transportation technologies, such as Driver Assistance Systems (DAS) [1], [2], Advanced Driver Assistance Systems (ADAS) [3], [4], [5], [6], and Autonomous Driving Systems (ADS) [7], [8], [9], [10], [11]. These systems play a pivotal role in enhancing road safety, improving the understanding of urban driving conditions, and monitoring traffic signs for maintenance and regulatory compliance. The output generated by TSR serves as a critical input for DAS, ADAS, ADS, and other intelligent systems, meaning the accuracy of TSR is crucial in shaping the decisions made by these systems. Inaccuracies in traffic sign recognition could lead to erroneous system responses, potentially resulting in severe consequences, including traffic accidents with fatal outcomes. Therefore, ensuring the reliability and precision of TSR is vital to minimizing risks and enhancing road safety across the board.

To enhance the recognition of traffic signs in Indonesia, a wide range of approaches and methods have been implemented, evolving from simple algorithms to more advanced techniques. Early methods include shape detection and distance calculation [12], [13], [14], which are computationally lightweight but may lack robustness in complex environments. Clustering techniques [15], have also been explored, enabling the grouping of similar traffic signs based on certain features. However, in recent years, more sophisticated methods have emerged, particularly in the domain of deep learning. One of the most prominent techniques is the use of Convolutional Neural Networks (CNN) [16], [17], [18], which have proven to be highly effective in image classification tasks. The CNN approach, due to its ability to automatically learn and extract features from raw images, has consistently outperformed traditional methods, achieving superior accuracy in traffic sign recognition tasks. Research has shown that the previous CNN models, when properly trained and optimized, can make them highly reliable for real-world applications [19], [20], [21], [22], [23].

Previous research in traffic sign recognition in Indonesia using CNN has demonstrated significant advancements in both accuracy and model performance. Initially, the model trained on 1,750 images and tested on 300 images achieved a solid performance with a loss value of 0.107 and an accuracy of 97.33% [24]. Building upon this foundation, further improvements were made by incorporating additional convolutional and subsampling layers, which enhanced the model's ability to capture finer details and patterns, reducing the loss to 0.0159 and boosting accuracy to an impressive 99.33% [25]. This was achieved using the same dataset, proving the model's robustness in improving performance through architectural modifications. The next step in the research introduced a more sophisticated multi-scale CNN architecture [26], where the outputs from each subsampling layer were directly utilized in the classification layer. This innovative approach allowed the model to capture features at different levels of abstraction, further refining its predictive capabilities. As a result, the model achieved an even lower loss value of 0.017 and an accuracy of 99.67%, outperforming all previous iterations [27].

In recent years, CNN has seen significant advancements, leading to the development of more versatile and complex architectures [28], [29], [30], [31], [32], [33], [34], [35]. There has been growing interest in CNN models that generate outputs for multiple tasks, a technique known as Multi-Task Learning (MTL) [36]. MTL allows a single CNN model to handle multiple related tasks simultaneously, sharing learned features across tasks. In security, for instance, MTL has been successfully applied in face detection and recognition systems [37], [38], improving accuracy by jointly learning features for both identification and facial expression recognition. In healthcare, MTL is used for predicting multiple conditions simultaneously, enabling more comprehensive diagnostics [39], [40], [41]. In transportation, MTL models have been applied to vehicle recognition, handling tasks such as type classification and license plate detection concurrently [42]. Furthermore, in agriculture, MTL is being used for tasks such as plant disease detection and yield prediction [43], demonstrating its versatility across diverse real-world applications. These

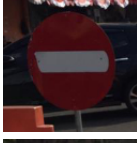
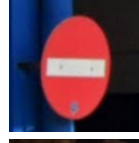
advancements in CNN and MTL continue to push the boundaries of what is possible in machine learning, enabling more efficient, accurate, and scalable solutions.

The proposed research seeks to take this progress even further by incorporating multi-scale CNN architectures [26], [27], within an MTL framework [42]. The use of a multi-scale approach can enhance the model's ability to recognize traffic signs in varied and complex environments. Additionally, the integration of multi-task learning will enable the model to handle multiple related tasks concurrently, sharing learned features across tasks. This work differs from previous studies by integrating multi-scale CNN with multi-task learning in a dataset of Indonesian traffic signs captured under various environmental conditions.

2. RESEARCH METHOD

The data collection process utilizes a dataset composed of 2,050 images across 10 distinct classes, derived from prior studies [24], [25], [27]. This dataset was collected in several ways, namely the first is direct capture with the help of a cellphone camera, the second is indirect capture by utilizing screenshots on a digital map application (Google Maps), the third is capture with several different angles, the fourth is capture done during the day and capture done at night. The colored image is the result of the dataset collection process, then the cropping process is carried out in the area of concern (region of interest/ROI), with the pixel size of the image from the cutting process being 128x128. Table I shows several examples of traffic sign image datasets that have been successfully collected. This dataset is then split into 1,750 images for training and 300 images for testing. Upon collection, the dataset undergoes a crucial preprocessing phase to ensure it is adequately prepared for input into the CNN model. Table I presents several examples of the successfully curated traffic sign image datasets, along with their corresponding classes.

Table 1. Model Training Results

No	Class	Example 1	Example 2	Example 3	No	Class	Example 1	Example 2	Example 3
1	Prohibition				6	Warning			
2					7				
3					8				
4					9				
5					10				

The dataset shown in the image consists of traffic signs categorized into two main classes: prohibition and warning. Each row in the table represents a specific type of traffic sign, accompanied by three example images per type, which illustrate the variety and diversity of these signs as they appear in real-world settings. This dataset is designed to support the multi-task learning CNN model by providing distinct examples for two classification tasks: identifying one of 10 specific traffic signs and determining whether a sign falls under the prohibition or warning

category. The first five rows belong to the "prohibition" class, which includes signs that convey rules or restrictions. Examples of these are "No Parking" signs, which vary in background and lighting, showing different appearances of the symbol. Other signs include the "Stop" sign, characterized by its octagonal red shape, the "No Entry" sign with a red circle and horizontal white bar, speed limit signs (e.g., 40 km/h) that vary in font style and context, and the "No U-Turn" sign, represented in different angles and conditions.

The last five rows are from the "warning" class, which consists of signs that alert drivers to potential hazards or important road conditions. These include signs warning of upcoming traffic lights, intersections, pedestrian or bicycle crossings, school zones, and general caution signs, often indicated by an exclamation mark. Each of these types is shown with variations in colors, shapes, and designs, reflecting regional differences and environmental factors. This dataset effectively captures the variability of traffic signs as they appear in different real-world scenarios, including changes in lighting, background, angle, and sign conditions, such as being faded or partially obstructed. This diversity is essential for training a robust CNN model, as it ensures that the model learns to recognize the key features of each sign type, even under varying conditions. By providing a range of examples, the dataset helps the model generalize well, enabling accurate recognition and classification of signs that it might not have encountered during training.

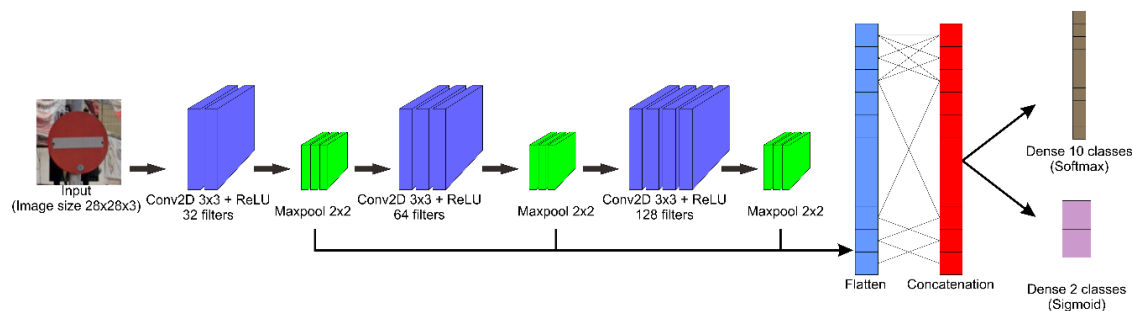


Figure 1. CNN architecture

Subsequently, the development of the CNN architecture takes center stage. This involves fine-tuning the model by modifying layer configurations, such as adjusting the number of convolutional and dense layers, integrating activation functions, applying normalization techniques, and performing parameter optimization. The CNN architecture proposed in this research can be shown in Figure 1.

The initial layer in a CNN is the convolutional layer, which performs a mathematical operation known as convolution, a process that combines two signals to produce a third signal. In the context of image processing, the input signal is typically a grayscale image, although color images can also be used. This input is processed through a filter, producing the output signal (the third signal) [44]. The number of filters employed is a tunable hyperparameter, often set to values such as 8, 16, 32, and so on, but it is not strictly limited. The size of the filters can vary, with common choices being 3x3, 5x5, 7x7, or larger, depending on the complexity of the task. In the proposed CNN architecture, as illustrated in Figure 1, three convolutional layers are implemented, with the number of filters for each convolutional layer being 32, 64, and 128, respectively, and the size of the filters used is 3x3. After passing through the convolutional layers, the output progresses to the next stage, the pooling layer. The primary objective of this layer is to achieve spatial invariance by reducing the dimensionality of the feature maps. This is accomplished by summarizing the neurons within small spatial neighborhoods. In this architecture, max-pooling is used, where the maximum value within each local neighborhood is selected [45]. A 2x2 window function is typically employed for this purpose, halving the spatial dimensions of the input feature map. As demonstrated in Figure 1, each convolutional layer is followed by a corresponding pooling layer, resulting in three pooling layers in the proposed CNN design.

The outputs from each pooling layer are subsequently passed to a fully connected layer. However, before this step, a flattening operation is performed, where the multidimensional output of the pooling layer is transformed into a one-dimensional vector, which serves as the input to the

fully-connected layer. Unlike traditional architectures where only the output of the final pooling layer is fed into the fully connected layer, in this design, the outputs of all pooling layers are utilized, as shown in Figure 1. After the feature maps are concatenated, the fully-connected layer is divided into two branches, each designed for a different classification task. The first branch utilizes the Softmax activation function, which outputs a probability distribution across 10 possible classes, each representing a specific type of traffic sign. This allows the model to classify the input image as one of the 10 traffic sign categories. The second branch employs the Sigmoid activation function, which outputs probabilities for two distinct classes: prohibition signs and warning signs. Sigmoid is suited for binary classification tasks. In the proposed model, categorical cross-entropy is used as a loss function. Categorical cross-entropy is a widely used loss function in multi-class classification problems, particularly when the model outputs a probability distribution across multiple classes [46], [47]. Once the architecture is refined, the model will be rigorously trained to achieve optimal performance, ensuring that the enhancements in the trained model translate to superior results.

Finally, the trained model is tested using 300 testing images, which are different from the training dataset. The performance evaluation of the proposed model was carried out using a confusion matrix and several metrics, namely accuracy, precision, recall, and F1-score. A confusion matrix is a widely used tool for evaluating the performance of classification algorithms, applicable to both binary and multiclass problems. It presents a cross-tabulated summary that highlights how accurately a model classifies instances and where misclassifications occur, thereby providing key insights into the predictive effectiveness of the model. By analyzing the distribution of correct and incorrect predictions, the confusion matrix serves as a fundamental metric for assessing the reliability and accuracy of classification outcomes [48], [49], [50], [51]. Confusion matrix visualization with a heatmap is used to identify misclassification patterns, while performance analysis provides detailed insight into the model's ability to distinguish each traffic sign and also classify them into prohibition signs or warning signs.

3. RESULTS AND DISCUSSION

Table 2. Model training results

epoch	CNN				MS-CNN			
	Output 1		Output 2		Output 1		Output 2	
	Loss	Accuracy (%)	Loss	Accuracy (%)	Loss	Accuracy (%)	Loss	Accuracy (%)
1	1,3827	48,17	0,1208	95,6	0,8869	71,6	0,0869	97,26
2	0,4271	89,89	0,0195	99,31	0,1552	97,14	0,0111	99,6
3	0,1648	96,34	0,0096	99,6	0,0685	98,74	0,0068	99,77
4	0,0997	97,77	0,0165	99,43	0,0393	99,43	0,0067	99,71
5	0,0693	98,29	0,0065	99,83	0,0193	99,66	0,0004	100
6	0,0498	98,69	0,0019	100	0,0167	99,71	0,0003	100
7	0,0323	99,2	0,0058	99,77	0,0119	99,83	0,0088	99,77
8	0,0256	99,31	0,0023	99,89	0,003	100	0,0001	100
9	0,0184	99,66	0,0026	99,94	0,0059	99,89	0,001	100
10	0,0162	99,49	0,0033	99,83	0,0011	100	0,00008	100
11	0,0057	100	0,0003	100	0,0005	100	0,00003	100
12	0,0097	99,83	0,0086	99,71	0,0067	99,89	0,0021	99,94
13	0,0084	99,77	0,0001	100	0,001	100	0,0014	99,94
14	0,0016	100	0,00005	100	0,0003	100	0,00009	100
15	0,0117	99,54	0,0004	100	0,0001	100	0,00002	100
16	0,0011	100	0,00002	100	0,00009	100	0,00001	100
17	0,0089	99,83	0,004	99,83	0,0003	100	0,000009	100

epoch	CNN				MS-CNN			
	Output 1		Output 2		Output 1		Output 2	
	Loss	Accuracy (%)	Loss	Accuracy (%)	Loss	Accuracy (%)	Loss	Accuracy (%)
18	0,0056	99,94	0,0037	99,89	0,0083	99,89	0,0027	99,89
19	0,0006	100	0,00009	100	0,00006	100	0,000008	100
20	0,0002	100	0,00003	100	0,00004	100	0,000005	100

Table 2 presents the training results of an ordinary CNN and a multi-scale CNN (MS-CNN) model in a multi-task learning setup. The model has been trained over 20 epochs using the Adam optimizer with a learning rate of 0.0002. The CNN is tasked with two outputs: classifying 10 different types of traffic signs and distinguishing between warning and prohibition signs. As shown in the table below, for the ordinary CNN, Output 1 starts with a high loss of 1.3827 and low accuracy of 48.17% in epoch 1, but the model gradually improves, reaching 100% accuracy by epoch 11 with a loss of 0.0057. Output 2 starts with a relatively high accuracy of 95.6% and lower loss of 0.1208, achieving 100% accuracy by epoch 6 with a minimal loss of 0.0019. Overall, the ordinary CNN demonstrates a steady improvement in accuracy and a reduction in loss for both outputs over time, with both outputs stabilizing at 100% accuracy by epoch 11 for Output 1 and epoch 6 for Output 2. In contrast, the MS-CNN achieves faster convergence and more stable results. Output 1 begins with a lower loss (0.8869) and higher initial accuracy (71.6%) compared to the ordinary CNN, reaching 100% accuracy by epoch 8 with a very low loss of 0.003. Output 2 has similarly high initial accuracy (97.26%) and a lower starting loss (0.0869) than the ordinary CNN. It achieves 100% accuracy by epoch 8, with the loss gradually decreasing to stabilize near zero by the final epochs. The MS-CNN shows superior performance in terms of convergence speed and stability. The multi-scale design of the MS-CNN enables it to achieve high accuracy faster than the ordinary CNN, likely due to its ability to capture multi-scale features effectively across its convolution layers. The training process of the MS-CNN model in MTL can be better and further understood by analyzing both the training metrics and the accompanying charts.

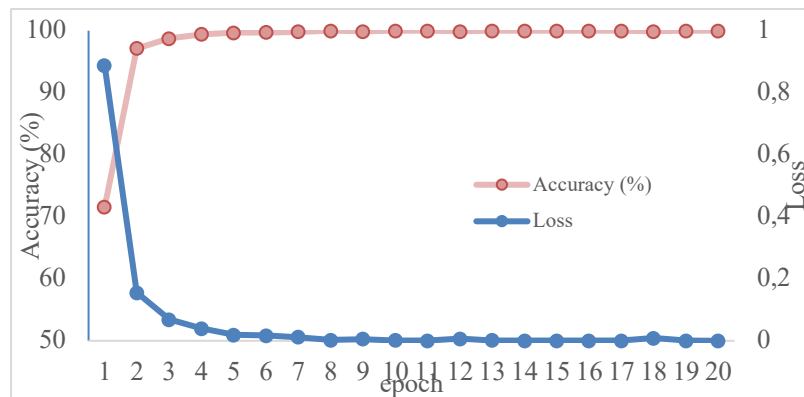


Figure 2. MS-CNN-MTL training results chart for Output 1

Figure 2 depicts the training behavior of the MS-CNN model for classifying 10 traffic signs. Initially, the model starts with a high loss of 0.8869 and an accuracy of 71.6% in the first epoch, suggesting that the model is in the early stages of learning to distinguish the different traffic signs. However, the chart reveals a steep drop in loss by the second epoch, down to 0.1552, while accuracy sharply rises to 97.14%. This rapid improvement indicates that the model quickly learns essential features, reflecting its ability to adapt. From epochs 3 to 5, there is a steady decline in loss, reaching as low as 0.0193 by the fifth epoch, and the accuracy continues to improve, almost achieving 100%. The loss curve flattens out around epoch 8, where the model reaches 100% accuracy for the first time, with a minimal loss of 0.003. This achievement marks the beginning of a phase where the model maintains perfect classification accuracy throughout the remaining epochs. The stability of both the loss and accuracy curves beyond epoch 8 suggests that the model

has effectively generalized the key features of the 10 traffic signs, allowing it to make consistently correct predictions.

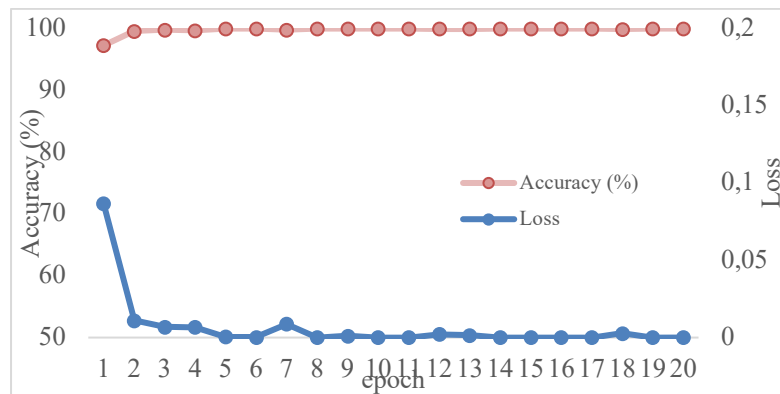


Figure 3. MS-CNN-MTL training results chart for Output 2

Figure 3 illustrates the training behavior of the MS-CNN model for categorizing signs as either warning or prohibition. The initial phase, as shown in the chart, starts with a lower loss of 0.0869 and a high accuracy of 97.26%. The loss drops sharply by the second epoch to 0.0111, and accuracy reaches 99.6%. This quick convergence, depicted in the steep drop of the loss curve and the rise of the accuracy curve, indicates that this task is relatively easier for the model, likely due to the broader classification categories. By the fifth epoch, the model has achieved 100% accuracy, maintaining this performance through most of the training. The loss values stabilize at very low levels, around 0.0014, indicating minimal classification errors. Although there are minor fluctuations observed in the twelfth and eighteenth epochs, where accuracy slightly dips to 99.94% and 99.89%, respectively, the model quickly recovers and returns to 100% accuracy. These minor variations, which are depicted as small dips in the accuracy curve, suggest that the model may have encountered slightly harder-to-classify samples but was resilient enough to adjust and stabilize. The charts visually represent the efficient learning dynamics of the multi-task CNN model. The rapid convergence seen in both outputs during the initial epochs showcases the effectiveness of the Adam optimizer and the chosen learning rate of 0.0002. For Output 1, the model refines its understanding over a few epochs, eventually stabilizing at perfect classification, while Output 2 reaches stable, high performance even faster. The overall smooth and stable lines of the accuracy and loss curves indicate a well-optimized training process, where the model consistently learns and improves. The training charts confirm the robustness and efficiency of the proposed model. The steep initial drops in loss and quick rise to high accuracy demonstrate rapid learning, while the stable maintenance of 100% accuracy across later epochs reflects the model's ability to generalize well.

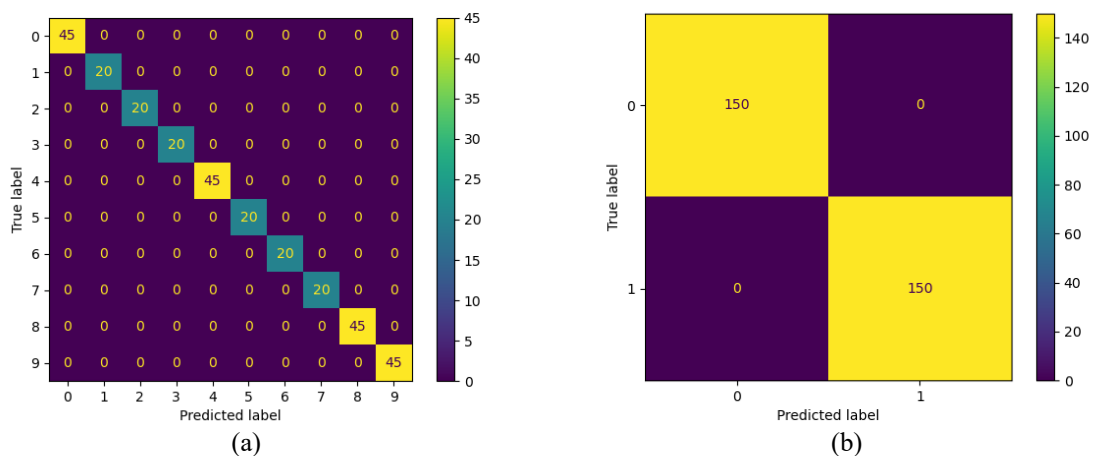


Figure 4. MS-CNN-MTL testing results' confusion matrix
(a) Output 1, (b) Output 2

The testing results for the multi-task learning CNN model are illustrated through two confusion matrices, each representing a different output task, as shown in Figure 4. The matrix on the left (a) corresponds to Output 1, which involves classifying 10 specific traffic signs (labeled 0 to 9). Each class has either 20 or 45 samples, as seen from the diagonal values, and all predictions fall precisely on the diagonal, with no off-diagonal entries. This indicates perfect classification accuracy for Output 1, as the model correctly identified each traffic sign across all samples. The matrix on the right (b) corresponds to Output 2, which involves categorizing traffic signs as either "warning" (class 0) or "prohibition" (class 1). Each category contains 150 samples, and the model again achieves perfect classification, with all predictions correctly placed on the diagonal and no misclassifications. This results in 100% accuracy for the binary classification task in Output 2. Overall, both matrices indicate that the model performs exceptionally well, achieving 100% accuracy on both tasks. This suggests that the multi-scale CNN architecture is highly effective for this multi-task learning application, providing robust and accurate performance across both outputs.

Table 3. Classification reports for Output 1

	Precision (%)	Recall (%)	F1-Score (%)	Support
0	100	100	100	45
1	100	100	100	20
2	100	100	100	20
3	100	100	100	20
4	100	100	100	45
5	100	100	100	20
6	100	100	100	20
7	100	100	100	20
8	100	100	100	45
9	100	100	100	45
Accuracy			100	300
Macro avg	100	100	100	300
Weighted avg	100	100	100	300

Table 4. Classification reports for Output 2

	Precision (%)	Recall (%)	F1-Score (%)	Support
0	100	100	100	150
1	100	100	100	150
Accuracy			100	300
Macro avg	100	100	100	300
Weighted avg	100	100	100	300

The performance evaluation of the proposed model for traffic sign recognition is comprehensively summarized using standard classification metrics derived from the confusion matrix, including precision, recall, F1-score, and accuracy. For Output 1, the results presented in Table 3 demonstrate that the model achieved perfect classification performance across all ten classes. Each class reported a precision, recall, and F1-score of 1.00, regardless of the sample size (support), which varied between 20 and 45 samples per class. The macro and weighted averages of these metrics also yielded perfect scores, and the overall accuracy for Output 1 is 100%. These results strongly suggest that the model is highly capable of distinguishing between various types of traffic signs without any misclassifications, even in the presence of an imbalanced distribution of sample sizes across classes. Similarly, Output 2, as shown in Table 4, indicates that the model also performed flawlessly in the binary classification task. Each of the two classes had equal representation, with 150 samples each, and the model achieved a precision, recall, and F1-score of 1.00 for both classes. The macro and weighted averages again reflect perfect performance, with the overall accuracy reported at 100%. This demonstrates the model's ability to effectively generalize and correctly categorize broader semantic groupings of traffic signs, which is essential for real-time decision-making in the implementation of traffic sign recognition.

4. CONCLUSION

This study presents a CNN-based approach to traffic sign recognition, specifically adapted for Indonesian traffic signs, utilizing a dataset from previous work. The proposed model

incorporates a Multi-Scale Convolutional Neural Network (MS-CNN) within a Multi-Task Learning (MTL) framework. The multi-scale architecture enhances the model's capacity to detect traffic signs under challenging visual conditions, while the MTL framework enables simultaneous learning of two related tasks: classifying traffic signs into specific types and categorizing them as either warning or prohibition signs. Training results demonstrated that the MS-CNN model outperformed a standard CNN baseline, achieving a lower initial loss, higher initial accuracy, and reaching 100% accuracy by the 8th epoch with a minimal error rate of 0.003. In the testing phase, the MS-CNN also exhibited exceptional performance. The confusion matrix results further validate this, showing perfect classification across both tasks. For the first output task (traffic sign type classification), the model achieved 100% precision, recall, and F1-score across all ten traffic sign classes. Similarly, for the second output task (traffic sign category classification), the model again demonstrated perfect performance, correctly identifying all signs as either warning or prohibition, with all evaluation metrics scoring 100. These results underscore the effectiveness of integrating multi-scale features with multi-task learning in traffic sign recognition. For future research, extending this approach by incorporating a multi-head architecture into the MS-CNN within the MTL framework is proposed. A multi-head design would allow the model to process multiple inputs such as images in different color spaces (red green blue or RGB, grayscale, or hue saturation value or HSV, etc), through separate input branches of the network. By integrating diverse representations of the same image, the network can leverage complementary information from different color spaces to improve its robustness and accuracy, particularly in complex or variable environmental conditions.

REFERENCES

- [1] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three Decades of Driver Assistance Systems: Review and Future Perspectives," *IEEE Intell. Transport. Syst. Mag.*, vol. 6, no. 4, pp. 6–22, 2014, doi: 10.1109/MITS.2014.2336271.
- [2] B. Reimer, "Driver Assistance Systems and the Transition to Automated Vehicles: A Path to Increase Older Adult Safety and Mobility?," *Publ. Pol. Aging. Rep.*, vol. 24, no. 1, pp. 27–31, Mar. 2014, doi: 10.1093/ppar/prt006.
- [3] F. Jiménez, J. E. Naranjo, J. J. Anaya, F. García, A. Ponz, and J. M. Armingol, "Advanced Driver Assistance System for Road Environments to Improve Safety and Efficiency," *Transportation Research Procedia*, vol. 14, pp. 2245–2254, 2016, doi: 10.1016/j.trpro.2016.05.240.
- [4] A. Ziebinski, R. Cupek, D. Grzechca, and L. Chruszczyk, "Review of advanced driver assistance systems (ADAS)," presented at the PROCEEDINGS OF THE INTERNATIONAL CONFERENCE OF COMPUTATIONAL METHODS IN SCIENCES AND ENGINEERING 2017 (ICCMSE-2017), Thessaloniki, Greece, 2017, p. 120002. doi: 10.1063/1.5012394.
- [5] V. K. Kukkala, J. Tunnell, S. Pasricha, and T. Bradley, "Advanced Driver-Assistance Systems: A Path Toward Autonomous Vehicles," *IEEE Consumer Electron. Mag.*, vol. 7, no. 5, pp. 18–25, Sep. 2018, doi: 10.1109/MCE.2018.2828440.
- [6] G. De-Las-Heras, J. Sánchez-Soriano, and E. Puertas, "Advanced Driver Assistance Systems (ADAS) Based on Machine Learning Techniques for the Detection and Transcription of Variable Message Signs on Roads," *Sensors*, vol. 21, no. 17, p. 5866, Aug. 2021, doi: 10.3390/s21175866.
- [7] J. Levinson *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany: IEEE, Jun. 2011, pp. 163–168. doi: 10.1109/IVS.2011.5940562.
- [8] K. Othman, "Exploring the implications of autonomous vehicles: a comprehensive review," *Innov. Infrastruct. Solut.*, vol. 7, no. 2, p. 165, Apr. 2022, doi: 10.1007/s41062-022-00763-6.
- [9] J. Zhao *et al.*, "Autonomous driving system: A comprehensive survey," *Expert Systems with Applications*, vol. 242, p. 122836, May 2024, doi: 10.1016/j.eswa.2023.122836.
- [10] B. Padmaja, Ch. V. K. N. S. N. Moorthy, N. Venkateswarulu, and M. M. Bala, "Exploration of issues, challenges and latest developments in autonomous cars," *J Big Data*, vol. 10, no. 1, p. 61, May 2023, doi: 10.1186/s40537-023-00701-y.

- [11] D. Garikapati and S. S. Shetiya, "Autonomous Vehicles: Evolution of Artificial Intelligence and the Current Industry Landscape," *BDCC*, vol. 8, no. 4, p. 42, Apr. 2024, doi: 10.3390/bdcc8040042.
- [12] O. R. Sitanggang, H. Fitriyah, and F. Utaminingrum, "Sistem Deteksi dan Pengenalan Jenis Rambu Lalu Lintas Menggunakan Metode Shape Detection Pada Raspberry Pi," *JPTIHK*, vol. 2, no. 12, pp. 6108–6117, Dec. 2018.
- [13] T. Oddy Chrisdwianto, H. Fitriyah, and E. Rosana Widasari, "Perancangan Sistem Deteksi dan Pengenalan Rambu Peringatan Menggunakan Metode Template Matching," *JPTIHK*, vol. 2, no. 3, pp. 1265–1274, Mar. 2018.
- [14] G. Romadhon and Murinto, "Aplikasi pengenalan citra rambu lalu lintas berbentuk lingkaran menggunakan metode jarak city-block," *JSTIE*, vol. 2, no. 2, pp. 286–294, Jun. 2014.
- [15] C. Rahmad, I. Fauziah Rahmah, and R. Andrie Asmara, "Deteksi dan pengenalan rambu lalu lintas di indonesia menggunakan RGBN dan Gabor," in *SENTRINOV*, 2017, pp. T113–22.
- [16] Y. LeCun *et al.*, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec. 1989, doi: 10.1162/neco.1989.1.4.541.
- [17] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object Recognition with Gradient-Based Learning," in *Shape, Contour and Grouping in Computer Vision*, vol. 1681, in Lecture Notes in Computer Science, vol. 1681, Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 319–345. doi: 10.1007/3-540-46805-6_19.
- [18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [19] H. H. Aghdam and E. J. Heravi, *Guide to Convolutional Neural Networks: A Practical Application to Traffic-Sign Detection and Classification*, Softcover reprint of the original 1st edition 2017. Cham: Springer International Publishing, 2017.
- [20] R. Venkatesan and B. Li, *Convolutional neural networks in visual computing: a concise guide*. Boca Raton, FL: CRC Press, 2018.
- [21] V. E. Balas, R. Kumar, and R. Srivastava, Eds., *Recent trends and advances in artificial intelligence and Internet of Things*. in Intelligent systems reference library, no. 172. Cham: Springer, 2020.
- [22] Y. Zhang, H. G. Soon, D. Ye, J. Y. H. Fuh, and K. Zhu, "Powder-Bed Fusion Process Monitoring by Machine Vision With Hybrid Convolutional Neural Networks," *IEEE Trans. Ind. Inf.*, vol. 16, no. 9, pp. 5769–5779, Sep. 2020, doi: 10.1109/TII.2019.2956078.
- [23] N. I. Chervyakov, P. A. Lyakhov, M. A. Deryabin, N. N. Nagornov, M. V. Valueva, and G. V. Valuev, "Residue Number System-Based Solution for Reducing the Hardware Cost of a Convolutional Neural Network," *Neurocomputing*, vol. 407, pp. 439–453, Sep. 2020, doi: 10.1016/j.neucom.2020.04.018.
- [24] M. Akbar, "Traffic sign recognition using convolutional neural networks," *J. Teknol. dan Sist. Komput.*, vol. 9, no. 2, pp. 120–125, Apr. 2021, doi: 10.14710/jtsiskom.2021.13959.
- [25] M. Akbar, "Pengenalan Rambu Lalu-lintas menggunakan Convolutional Neural Network (Studi Kasus: Rambu Lalu-lintas Indonesia)," *InfoTekJar*, vol. 6, no. 2, pp. 272–276, Mar. 2022, doi: <http://dx.doi.org/10.30743/infotekjar.v6i2.4564>.
- [26] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale Convolutional Networks," in *The 2011 International Joint Conference on Neural Networks*, San Jose, CA, USA: IEEE, Jul. 2011, pp. 2809–2813. doi: 10.1109/IJCNN.2011.6033589.
- [27] M. Akbar, A. S. Purnomo, and S. Supatman, "Multi-Scale Convolutional Networks untuk Pengenalan Rambu Lalu Lintas di Indonesia," *SISFOKOM*, vol. 11, no. 3, pp. 310–315, Dec. 2022, doi: 10.32736/sisfokom.v11i3.1452.
- [28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
- [29] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," 2013, *arXiv*. doi: 10.48550/ARXIV.1311.2901.
- [30] C. Szegedy *et al.*, "Going Deeper with Convolutions," 2014, *arXiv*. doi: 10.48550/ARXIV.1409.4842.

- [31] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2014, *arXiv*. doi: 10.48550/ARXIV.1409.1556.
- [32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2015, *arXiv*. doi: 10.48550/ARXIV.1512.03385.
- [33] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *AAAI*, vol. 31, no. 1, Feb. 2017, doi: 10.1609/aaai.v31i1.11231.
- [34] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," 2016, *arXiv*. doi: 10.48550/ARXIV.1608.06993.
- [35] S. Ramaneswaran, K. Srinivasan, P. M. D. R. Vincent, and C.-Y. Chang, "Hybrid Inception v3 XGBoost Model for Acute Lymphoblastic Leukemia Classification," *Computational and Mathematical Methods in Medicine*, vol. 2021, pp. 1–10, Jul. 2021, doi: 10.1155/2021/2577375.
- [36] S. Ruder, "An Overview of Multi-Task Learning in Deep Neural Networks," Jun. 15, 2017, *arXiv*: arXiv:1706.05098. Accessed: Feb. 13, 2024. [Online]. Available: <http://arxiv.org/abs/1706.05098>
- [37] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial Landmark Detection by Deep Multi-task Learning," in *Computer Vision – ECCV 2014*, vol. 8694, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., in Lecture Notes in Computer Science, vol. 8694. , Cham: Springer International Publishing, 2014, pp. 94–108. doi: 10.1007/978-3-319-10599-4_7.
- [38] X. Yin and X. Liu, "Multi-Task Convolutional Neural Network for Pose-Invariant Face Recognition," *IEEE Trans. on Image Process.*, vol. 27, no. 2, pp. 964–975, Feb. 2018, doi: 10.1109/TIP.2017.2765830.
- [39] Z. Wang *et al.*, "A Multi-Task Convolutional Neural Network for Lesion Region Segmentation and Classification of Non-Small Cell Lung Carcinoma," *Diagnostics*, vol. 12, no. 8, p. 1849, Jul. 2022, doi: 10.3390/diagnostics12081849.
- [40] S. Starke *et al.*, "Multitask Learning with Convolutional Neural Networks and Vision Transformers Can Improve Outcome Prediction for Head and Neck Cancer Patients," *Cancers*, vol. 15, no. 19, p. 4897, Oct. 2023, doi: 10.3390/cancers15194897.
- [41] B. Oliveira *et al.*, "A multi-task convolutional neural network for classification and segmentation of chronic venous disorders," *Sci Rep*, vol. 13, no. 1, p. 761, Jan. 2023, doi: 10.1038/s41598-022-27089-8.
- [42] D. Avianto, A. Harjoko, and Afiahayati, "CNN-Based Classification for Highly Similar Vehicle Model Using Multi-Task Learning," *J. Imaging*, vol. 8, no. 11, p. 293, Oct. 2022, doi: 10.3390/jimaging8110293.
- [43] Y. H. Kim and K. R. Park, "MTS-CNN: Multi-task semantic segmentation-convolutional neural network for detecting crops and weeds," *Computers and Electronics in Agriculture*, vol. 199, p. 107146, Aug. 2022, doi: 10.1016/j.compag.2022.107146.
- [44] S. W. Smith, *The scientist and engineer's guide to digital signal processing*. San Diego, Calif.: California Technical Pub., 1999.
- [45] D. Hutchison *et al.*, "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition," in *Artificial Neural Networks – ICANN 2010*, vol. 6354, K. Diamantaras, W. Duch, and L. S. Iliadis, Eds., in Lecture Notes in Computer Science, vol. 6354. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 92–101. doi: 10.1007/978-3-642-15825-4_10.
- [46] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Applied Soft Computing*, vol. 70, pp. 41–65, Sep. 2018, doi: 10.1016/j.asoc.2018.05.018.
- [47] Y. Chen, L. Li, W. Li, Q. Guo, Z. Du, and Z. Xu, "Fundamentals of neural networks," in *AI Computing Systems*, Elsevier, 2024, pp. 17–51. doi: 10.1016/B978-0-32-395399-3.00008-1.
- [48] A. Kulkarni, D. Chong, and F. A. Batarseh, "Foundations of data imbalance and solutions for a data democracy," in *Data Democracy*, Elsevier, 2020, pp. 83–106. doi: 10.1016/B978-0-12-818366-3.00005-8.

- [49] P. Singh, N. Singh, K. K. Singh, and A. Singh, “Diagnosing of disease using machine learning,” in *Machine Learning and the Internet of Medical Things in Healthcare*, Elsevier, 2021, pp. 89–111. doi: 10.1016/B978-0-12-821229-5.00003-3.
- [50] A. A. Lawan, N. Cavus, R. Yunusa, U. I. Abdulrazak, and S. Tahir, “Fundamentals of machine-learning modeling for behavioral screening and diagnosis of autism spectrum disorder,” in *Neural Engineering Techniques for Autism Spectrum Disorder, Volume 2*, Elsevier, 2023, pp. 253–268. doi: 10.1016/B978-0-12-824421-0.00020-5.
- [51] S. Sathyanarayanan, “Confusion Matrix-Based Performance Evaluation Metrics,” *AJBR*, pp. 4023–4031, Nov. 2024, doi: 10.53555/AJBR.v27i4S.4345.