Sequential Model-based Optimization Approach Deep Learning Model for Classification of Multi-class Traffic Sign Images

Si Thu Aung[1], Jartuwat Rajruangrabin[2], Ekkarut Viyanit[3]

Department of Mathematics, University at Buffalo, State University of New York, Buffalo, USA[1]
Rail and Modern Transports Research Center, National Science and Technology Development Agency (NSTDA), Thailand Science Park, Pathum Thani, Thailand[2]
Rail and Modern Transports Research Center, National Science and Technology Development Agency (NSTDA) Thailand Science Park, Pathum Thani, Thailand[3]

***Abstract*—Autonomous vehicles are currently gaining popularity in the future mobility ecosystem. The development of autonomous driving systems is still challenging in the research area of image processing and signal processing. Extensive research work was conducted on various traffic sign datasets. It achieved respectable results, but a robust network structure is still needed to develop to improve the traffic sign recognition (TSR) system. In this research work, there is an alternative approach to designing deep learning models, which are implemented in TSR systems. The proposed deep learning model was also tested with different datasets to obtain the generalized model. The proposed model was based on a convolutional neural network (CNN), and Bayesian Optimization optimizes the model's hyperparameters to find the best hyperparameters grid. After that, the optimized CNN model was used to classify the traffic sign images from three different datasets, including the German traffic sign recognition benchmark (GTSRB), the Belgium traffic sign classification (BTSC) dataset, and the Chinese traffic sign database, achieving the average accuracy scores of 99.57%, 99.15%, and 99.35%, respectively.**

***Keywords*—Autonomous driving; convolutional neural network; deep learning; traffic sign; optimization**

I. Introduction

Road traffic signs are fundamental and the most frequent visual tools to control traffic on the road, which are also used to communicate information to road users [1]. Thus, self-driving technologies use this road traffic information to control and drive an autonomous vehicle. For this reason, traffic sign recognition (TSR) systems have become quite popular and have been adopted in many engineering fields over the last decade [2]. Many applications are driven by this TSR system, such as advanced driver assistance systems (ADAS) [3], autonomous driving systems [4], and mobile mapping [5]. Moreover, many proposed problems related to the TSR system have been studied in recent research. In previous research [6], authors proposed two different models to identify factors affecting the retro-reflectivity of traffic signs. However, some authors found an overview of the interaction between road infrastructure and automated driving systems [7]. Additionally, the proposed problem of traffic sign recognition system problem has been developed by different approaches such as color-based, shape-based, and deep learning-based methods [8]. The deep learning-based method was the most popular because of its speed and accuracy [9].

Convolutional neural networks (CNN), one of the deep learning methods, can adjust the data without making any explicit specification, and CNN can also learn features efficiently from many samples without any preprocessing [8][9]. Some researchers applied CNN model to detect and recognize Thai traffic signs and achieved an average precision score of 0.93 [10]. An end-to-end convolutional network inspired by YOLOv2 was proposed and evaluated according to their Chinese traffic sign dataset (CTSD) [11]. Their approach achieved 90.37% in recall and 95.31% in precision, with the fastest detection speed of 0.017s per image. CNN was used to develop an autonomous traffic and road sign detection and recognition system of traffic signs collected from roads in Saudi Arabia, and there were 24 different traffic sign labels [11]. Their experimental results showed an accuracy of 100% in their dataset.

However, for the TSR system, a new architecture for neural network structure remains challenging for all the researchers in this field. Although many network architectures have been proposed, a robust network structure is still needed to develop for improving the TSR system. This research work implements an alternative approach to designing deep learning models for TSR systems. The main contribution of the proposed method is as follows:

- A new model is based on a convolutional neural network (CNN), and Bayesian Optimization optimizes the model's hyperparameters to find the best hyperparameters grid, and
- The proposed model is also tested with three different datasets to get the generalized model.

II. Literature Background

The performance comparison between machine learning algorithms and humans to classify road traffic signs and the results are reported in their research work [12]. The human performance of classification of the road sign achieved an accuracy of 99.22%, while the best machine learning approach, CNN, with 99.46% correct classification accuracy. According to their research, the performance of the CNN outperformed the human ability, and therefore CNN-based TSR system can be one of the best solutions to assist the driver. A hinge loss stochastic gradient descent (HLSGD) method was suggested to train CNN, and their CNN model was evaluated on the GTSRB [13]. This experiment showed a state-of-the-art result of 99.65% in recognition rate. In [14], an extremely fast detection module was proposed, and this module is based on a support vector machine (SVM) and CNN to detect and classify traffic signs. The overall classification accuracy of 98.24% for the German Traffic Sign Detection Benchmark (GTSDB) and 98.77% for CTSD. OneCNN, a single CNN model, was proposed and tested over multiple traffic sign datasets from Germany, Belgium, and Croatia [15]. Their model got a classification rate of 99.04% in the German dataset, 97.66% in the Belgium dataset, and 99.37% in the Croatia dataset.

Their research proposed a deep learning approach to traffic sign recognition systems, and their classification experiments were conducted over different traffic sign datasets from Germany and Belgium [8]. The recognition rate of their proposed model got an accuracy of 99.71% in the German traffic sign dataset and 98.87% in the Belgium traffic sign dataset. Some researchers have designed a detector that was implemented using an R-convolutional neural network, and the structure of MobileNet and CNN was also used to be the classifier of traffic signs [16]. They got a detection accuracy of 96% in prohibitory, 100% in mandatory, 100% in Danger, and 99% in unique on GTSDB. Their classification accuracy was 99.66% in GTSRB. MicronNet, a highly compact deep convolutional neural network for real-time embedded traffic sign recognition system, was based on the microarchitecture design principle proposed and achieved an accuracy of 98.9% on the German traffic sign dataset [17]. Recently, a novel neural network architecture called Separating-Illumination Network (Sill-Net) was proposed by Zhang et al. [18]. Their model got a classification accuracy of 99.68% on GTSRB. The implementation of TSR using YOLOv5 was conducted with r own dataset, and they have a performance measure of 97.70% for all classes [19]. Recently, a recognition system based on Faster R-CNN and YOLOv4 network was implemented and achieved an accuracy score of 99.20% [20]. Recent advancements in computer vision and machine learning greatly help improve the accuracy of the TSR system [21]. However, the proposed deep learning model has achieved better performance across three traffic sign datasets.

III. Materials and Methods

In this research work, three different traffic sign datasets used in testing the performance of the proposed CNN model are highly imbalanced. These three datasets are as follows:

- German traffic sign recognition benchmark,
- Belgium traffic sign classification dataset, and
- Chinese traffic sign database.

A. *German Traffic Sign Recognition Benchmark (GTSRB)*

In this dataset, there are single images with a multi-class classification problem of more than 40 traffic sign classes and more than 50,000 images, as shown in Fig. 1. Its ground truth data is reliable due to its semi-automatic annotation. The traffic sign images are raw RGB images with sizes ranging from 15×15 to 250×250 pixels [22].

Fig. 1. Dataset distribution of German traffic sign images.

B. *Belgium Traffic Sign Classification (BTSC) Dataset*

The Belgian traffic sign classification dataset (BTSC) [23] has 4533 training images and 2562 validation images split into 62 traffic sign types, as shown in Fig. 2. Compared with the GTSRB dataset, this dataset has different traffic sign pictograms, lighting conditions, occlusions, image resolutions, etc. Moreover, it contains categories that cluster different types of traffic signs (e.g., 50-speed-limit sign and 70-speed-limit sign), thereby raising the difficulty of the recognition task.

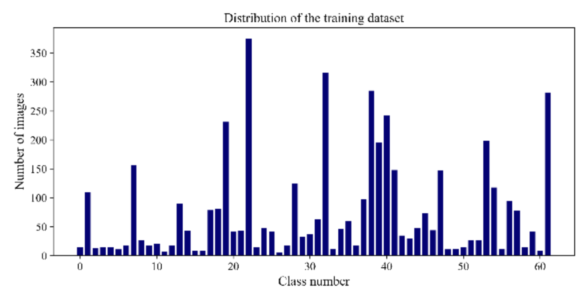


Fig. 2. Dataset distribution of Belgium traffic sign images.

C. *Chinese Traffic Sign Database*

This database [24] includes 6164 traffic sign images containing 58 sign categories, as shown in Fig. 3. The images are divided into two sub-databases as training database and testing database. The training database includes 4170 images, while the testing contains 1994 images. All images are annotated with the four coordinates of the sign and the category.

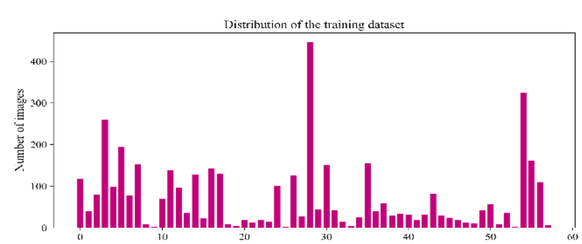


Fig. 3. Dataset distribution of Chinese traffic sign images.

D. Convolutional Neural Network (CNN)

Convolutional neural network (CNN) is one of the self-adaptive methods that can adjust to the data without any explicit specification [8]. CNN can also effectively learn features from numerous samples with minimal preprocessing [9]. Therefore, CNN was used in this research work to implement a traffic sign recognition system; the overall block diagram is shown in Fig. 4.

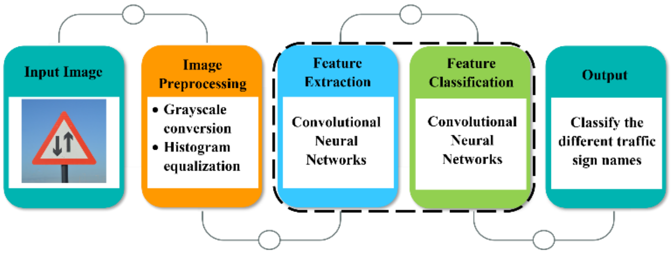


Fig. 4. Overall block diagram of traffic sign recognition system.

First, the image was loaded into the CNN model before being converted into a grayscale image with histogram equalization. Then, the images were rescaled by dividing with a maximum pixel value of 255. OpenCV library was used for these image preprocessing steps. During the preprocessing stage, all samples from GTSRB were down-samples or up-sample to 32×32, while the samples from BTSC and Chinese datasets were down-sample to 50×50 pixels. After these steps, the proposed CNN model applied feature extraction and classification of the image. The based CNN model with seven layers was inspired by LeNet-based architecture [25], and then the model was optimized by Bayesian optimization.

The proposed CNN used Keras with TensorFlow backend, and the model was trained and evaluated the classifier on all three datasets. The proposed model comprises several layers, including a convolutional layer, Rectified Linear Units (ReLU) [26], max pooling [27], and fully connected layers. The convolution and pooling layers were used as a feature extractor of the input images, and then fully connected layers were attached to flatten the matrix into a vector. Finally, softmax was used as an activation function to classify the outputs of different traffic sign labels. The softmax function in this research uses the categorical cross-entropy loss function [8]. The LeNet-based CNN architecture for the traffic sign recognition system is shown in Fig. 5.

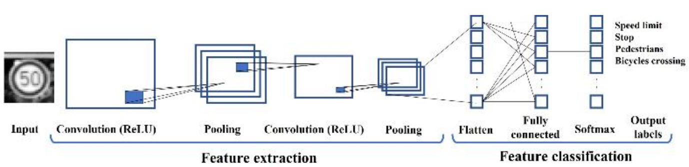


Fig. 5. LeNet-based CNN architecture for traffic sign recognition.

E. Hyperparameter Tuning Using Bayesian Optimization

In this research, the proposed CNN model was optimized by Bayesian Optimization, a sequential approach for the global optimization of black-box objective functions that are costly to evaluate [28]. The global maximizer or minimizer is mathematically needed to find an objective function [29]. The function *f* is defined as:

$$x^{*} = \arg\max_{x} f(x), \tag{1}$$

where x are the hyperparameters. In this research work, the set of hyperparameters was optimized to get the minimum value of the negative accuracy. Therefore, Bayesian optimization with a scikit-optimizer was used to find the minimum value. It provides a utility function to create the range of values to examine for each hyperparameter [30]. The following hyperparameters were optimized by scikit-optimize:

- the learning rate,
- the number of fully connected Dense layers,
- the number of nodes for each of the dense layers, and
- activation function.

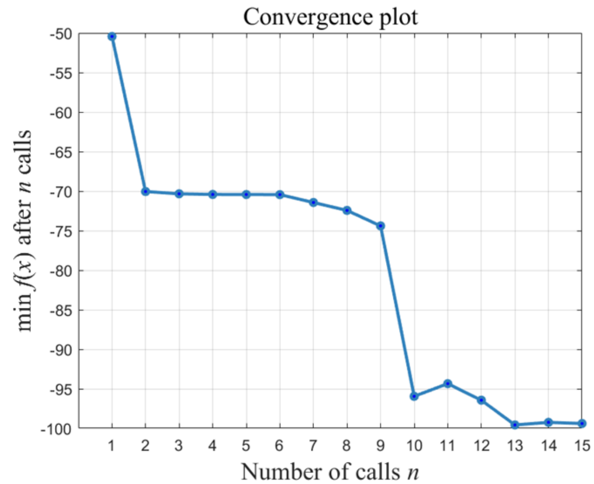


Fig. 6. The convergence plot of function min f(x).

Table I shows the range grid of the hyperparameters in this experiment. During the optimization process, there are 15 subsequent evaluations of f(x), and the target minimum value of accuracy was obtained in the 13th number of calls. In Fig. 6, the result can be visually seen with the convergence plot, and the test was done using the dataset of GTSRB. Therefore, Bayesian optimization [31] can give a better model performance on this dataset, and the optimized model will be used for testing with another two datasets. The selected hyperparameters combination is described in Table I.

TABLE I. HYPERPARAMETER GRID FOR DEEP LEARNING MODEL

Parameters	Range of grid
Learning rate	low=1e-6, high = 1e-2, prior= 'log-uniform', name = 'learning_rate'
Number of dense layers	low = 5, high = 15, name = 'num_dense_layer'
Number of dense nodes	low = 512, high = 1027, name = 'num_dense_nodes'

TABLE II. HYPERPARAMETER TUNING FOR DEEP LEARNING MODEL

Learning rate	Number of dense layers	Number of dense nodes	Activation function	Accuracy	loss
1.00E-05	5	512	relu	50.44	1.077
2.40E-04	13	954	sigmoid	70.05	3.564
3.10E-04	9	665	relu	70.34	0.2453
1.20E-05	10	930	relu	70.42	1.1444
3.70E-05	13	686	sigmoid	70.43	3.5517
3.00E-05	15	584	sigmoid	70.44	3.5515
7.80E-05	13	780	sigmoid	71.44	3.5543
7.60E-04	11	789	sigmoid	72.44	3.5674
2.70E-06	10	608	sigmoid	74.4	3.5515
7.30E-06	6	679	relu	95.95	1.2551
7.70E-06	9	977	relu	94.35	1.0966
1.00E-02	5	512	relu	96.44	3.5572
1.50E-04	**5**	**512**	**relu**	**99.57**	**0.2068**
1.60E-04	5	512	relu	99.24	0.5398
2.20E-04	5	1027	relu	99.39	0.2104

TABLE III. THE PROPOSED CNN ARCHITECTURE

Layers	Type	Output shape	Param#
1	Convolutional (ReLU)	(None, 28, 28, 60)	1560
2	Max-Pooling	(None, 14, 14, 60)	0
3	Convolutional (ReLU)	(None, 12, 12, 30)	16230
4	Max-Pooling	(None, 6, 6, 30)	0
5	Flatten	(None, 1080)	0
6	Fully connected (ReLU)	(None, 512)	553472
7	Fully connected (ReLU)	(None, 512)	262656
8	Fully connected (ReLU)	(None, 512)	262656
9	Fully connected (ReLU)	(None, 512)	262656
10	Fully connected (ReLU)	(None, 512)	262656
11	Dropout	(None, 512)	0
12	Fully connected (ReLU)	(None, 512)	22059
Total params: 1,643,945			
Trainable params: 1,643,945			
Non-trainable params: 0			

The detailed values of each hyperparameter can be seen in Table II, and as mentioned before, the best hyperparameter grid was in the 13th iteration with 99.57% accuracy and 0.2068 in loss value. Therefore, the proposed model was updated with this parameter grid and shown in Fig. 7. For training the GTSRB, the updated model architecture of convolutional layers, ReLU, max pooling, and fully connected layers are reported in Table III. The detailed structure of the proposed CNN model is shown in Fig. 7.

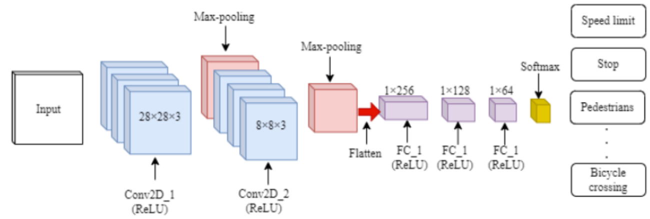
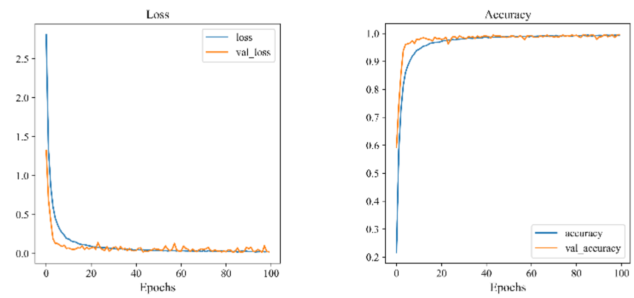


Fig. 7. The updated CNN architecture for classification of traffic sign images.

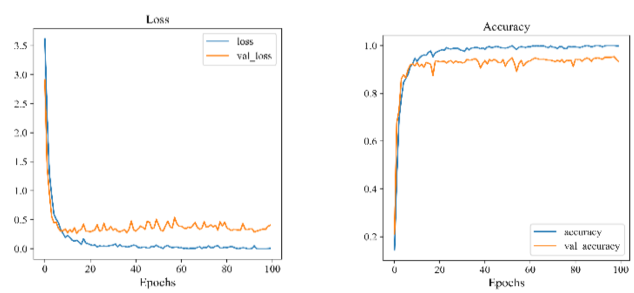
IV. RESULTS AND DISCUSSION

All experiments of CNN architecture and Bayesian optimization were done on a computer with an Intel Core i5 – 7300HQ CPU, 16 GB of RAM, and an NVIDIA GeForce GTX 1050 discrete GPU with 2 GB of RAM; this computer was enough to train and run models. Training and validation sets are already included in the datasets since they can be downloaded from the source link. Therefore, the developed CNN model was trained with a training set and evaluated with the model with a validation set. Fig. 8, Fig. 9, and Fig. 10 show the values of training loss and accuracy from different datasets and are compared with the values of validation loss and accuracy on those datasets. The training time per epoch of the proposed CNN model is 15.76 seconds, and the simulation ran 100 epochs per dataset. Our proposed CNN model achieved better accuracy with lower loss values on all three datasets.



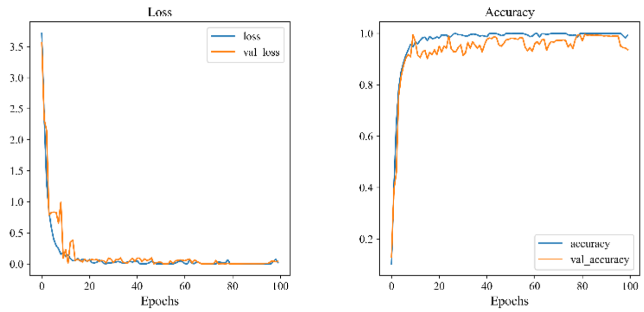
(a) (b)

Fig. 8. Comparison between training and validation of German traffic sign dataset regarding (a) loss and (b) accuracy.



(a) (b)

Fig. 9. Comparison between training and validation of Belgium traffic sign dataset in terms of (a) loss and (b) accuracy.



(a) (b)

Fig. 10. Comparison between training and validation of Chinese traffic sign dataset regarding (a) loss and (b) accuracy.

Performance is evaluated based on the GTSRB dataset by calculating precision, recall, F1-score, and accuracy:

$$\text{Precision }(\%)=\frac{\text{TP}}{\text{TP}+\text{FP}}\times 100, \quad (2)$$

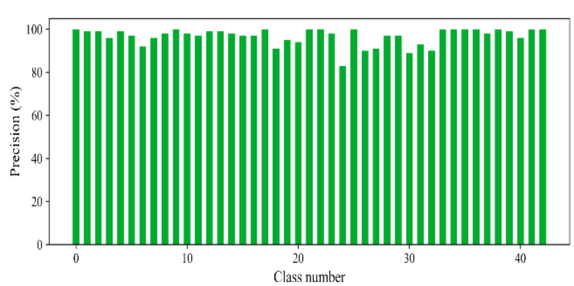
$$\text{Recall }(\%)=\frac{\text{TP}}{\text{TP}+\text{FN}}\times 100, \quad (3)$$

$$\text{F1-score }(\%)=\frac{2\times\text{TP}}{2\times(\text{TP}+\text{FP}+\text{FN})}\times 100, \quad (4)$$

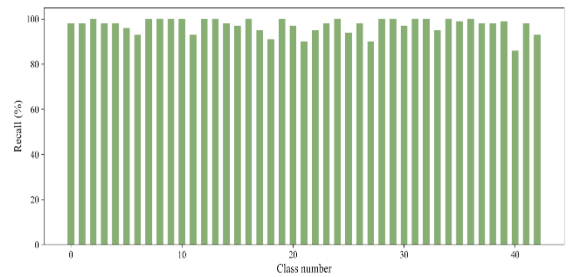
$$\text{Accuracy }(\%)=\frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{TN}+\text{FN}}\times 100, \quad (5)$$

Where TN is the number of true negatives, TP is the number of true positives, and FN and FP are the number of false negatives and false positives, respectively [32]. The precision, recall, and f1-score are calculated for individual labels to classify traffic signs.

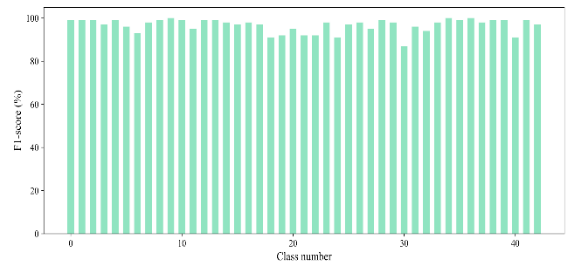
It clearly shows that the average percentage of the classification of 43 sign labels from the GTSRB dataset is 96.79% in precision measure, 97.21% in recall measure, and 96.65% in F1-score, as shown in Fig. 11. The proposed model is capable of classification of 61 different annotated sign labels and the performance measures of Belgium traffic sign database are shown in Fig. 12. The proposed model achieved 94.06% of precision measure, 93.82% of recall measure and 92.77% of F1-score. Moreover, the proposed model was also tested with another traffic database of the Chinese traffic sign database, and the average results on each performance measure were 91.24%, 86.11%, and 86.41%, respectively. In this database, there are 57 different sign labels; these performance measures can be seen in Fig. 13.



(a)

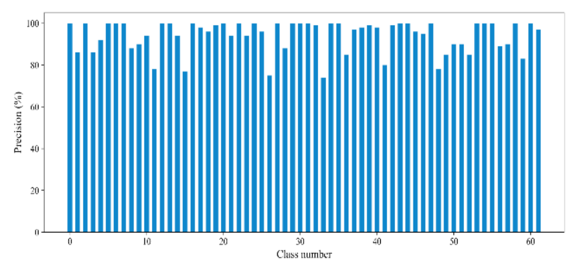


(b)

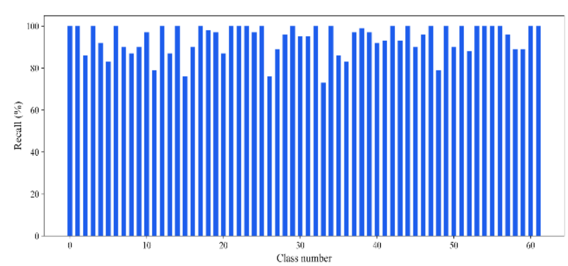


(c)

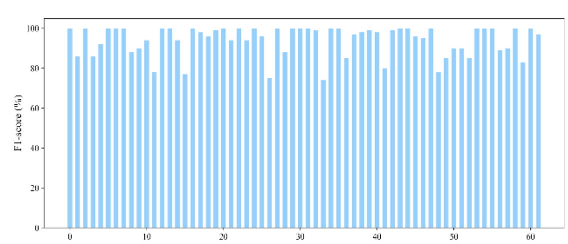
Fig. 11. Performance measures of German traffic sign dataset (a) precision, (b) recall, and (c) F1-score.



(a)

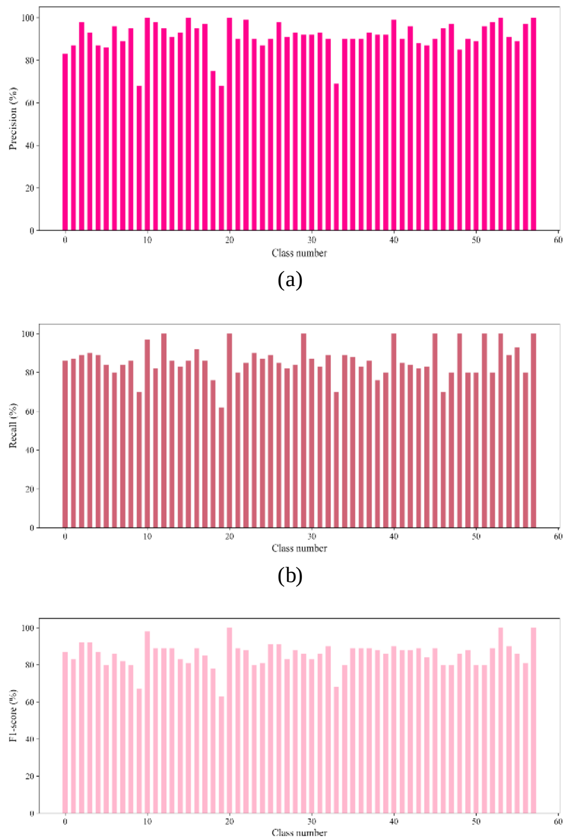


(b)



(c)

Fig. 12. Performance measures of Belgium traffic sign dataset (a) precision, (b) recall, and (c) F1-score.



(a) (b) (c)

Fig. 13. Performance measures of Chinese traffic sign dataset (a) precision, (b) recall, and (c) F1-score.

The new CNN model was proposed in this research work, and the experimental results are shown in the above section. In this section, a detailed discussion of the accuracy results from this proposed model and the existing deep learning models. The classification accuracies of the proposed model on the German, Belgium, and Chinese traffic sign datasets are mentioned as follows: the average accuracy of 99.57% on the German dataset, 99.15% on the Belgium dataset, and 99.35% on the Chinese dataset. A comparison between the recent research and the proposed CNN model on the German dataset is shown in Table IV. The proposed model achieved a better accuracy score than the previous study with CNN [15], and MicronNet [17], but the score is slightly less than CNN with three spatial transformer layers and SGD without momentum as the loss function optimizer [8]. However, our CNN model's total number of parameters is 1,643,945, which is much lower than their model parameters. (See Table VI). In contrast, the proposed model got the best accuracy value on the Belgium traffic dataset compared with the recent research, including CNN with three Spatial Transformer Networks [8], a Gaussian-Bernoulli deep Boltzmann machine-based (GDBM) hierarchical classifier [30], and other approaches [2]. The accuracy comparison between our proposed CNN model and the recent studies is shown in Table V. Therefore, the light weight CNN model with higher accuracy measures was successfully proposed in this research and can be used in real-time scenarios.

TABLE IV. ACCURACY COMPARISON OF GERMAN TRAFFIC SIGN DATASET BETWEEN THE PROPOSED MODEL AND RECENT STUDIES

References	Methods	Accuracy
[8]	CNN with 3 Spatial Transformer Networks	99.71%
[18]	Separating-Illumination Network (Sill-Net)	99.68%
Proposed model	Bayesian Optimization approach CNN model	99.57%
[15]	CNN	99.11%
[17]	MicronNet	98.9%

TABLE V. ACCURACY COMPARISON OF BELGIUM TRAFFIC SIGN DATASET BETWEEN THE PROPOSED MODEL AND RECENT STUDIES

References	Methods	Accuracy
Proposed model	Bayesian Optimization approach CNN model	99.15%
[33]	GDBM	98.92%
[8]	CNN with 3 Spatial Transformer Networks	98.87%
[15]	CNN	98.17%
[2]	INNLP+SCR(PI)	97.83%

TABLE VI. COMPARISON OF A NUMBER OF TRAINABLE PARAMETERS OF THE PROPOSED CNN MODEL WITH RECENT STUDIES

References	Methods	Trainable parameters
[8]	CNN with 3 Spatial Transformer Networks	14,629,801
Proposed model	Bayesian Optimization approach CNN model	1,643,945

V. CONCLUSION

In this paper, sequential model-based optimization (Bayesian optimization with scikit-optimize) approach deep learning model was proposed and tested with different traffic sign datasets, including GTSRB, BTSC, and the Chinese traffic sign database. The proposed model was in a rank with the top result in BTSC and the top three results in GTSRB. It is possible to show the high-quality results obtained from the proposed models using the Chinese dataset. Therefore, the proposed CNN model significantly improves traffic sign recognition (TSR) system performance compared with other existing studies. Further work should be conducted with other traffic sign datasets from different countries to guarantee the quality and accuracy levels of the proposed CNN model while comparing the real-world scenarios.

AUTHORS' CONTRIBUTIONS

Si Thu Aung conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the paper and approved the final draft. Jartuwat Rajruangrabin conceived and designed the experiments, analyzed the data, performed the computation work, authored, or reviewed drafts of the paper, and approved the final draft. Ekkarut Viyanit authored, or reviewed drafts of the paper, and approved the final draft.

FUNDING

The project was funded by "Program Management Unit for Competitiveness Enhancement (PMU-C), the Office of National Higher Education Science Research and Innovation Policy Council (NXPO)" under contract number C10F630211.

DATA AVAILABILITY STATEMENT

The German traffic sign recognition Benchmark (GTSRB) can be downloaded from https://benchmark.ini.rub.de/gtsrb_dataset.html Belgium traffic sign classification (BTSC) dataset can be obtained from https://btsd.ethz.ch/shareddata/ Chinese traffic sign database can be accessed from http://www.nlpr.ia.ac.cn/pal/trafficdata/recognition.html All datasets have been used for this study.

REFERENCES

[1] M. Koyuncu and S. Amado, "Effects of stimulus type, duration and location on priming of road signs: Implications for driving," Transportation Research Part F: Traffic Psychology and Behaviour, vol. 11, no. 2, pp. 108-125, 2008.

[2] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, "Traffic sign recognition—How far are we from the solution?," in The 2013 international joint conference on Neural networks (IJCNN), 2013: IEEE, pp. 1-8.

[3] S. B. Wali et al., "Vision-based traffic sign detectionand recognition systems: Current trends and challenges," Sensors, vol. 19, no. 9, p. 2093, 2019.

[4] J. Levinson et al., "Towards fully autonomous driving: Systems and algorithms," in 2011 IEEE intelligent vehicles symposium (IV), 2011: IEEE, pp. 163-168.

[5] R. Timofte, K. Zimmermann, and L. Van Gool, "Multi-view traffic sign detection, recognition, and 3D localisation," Machine vision and applications, vol. 25, pp. 633-647, 2014.

[6] R. Saleh, H. Fleyeh, and M. Alam, "An Analysis of the Factors Influencing the Retroreflectivity Performance of In-Service Road Traffic Signs," Applied Sciences, vol. 12, no. 5, p. 2413, 2022.

[7] T. Mihalj, H. Li, D. Babić, C. Lex, M. Jeudy, G. Zovak, D. Babić, A. Eichberger, "Road Infrastructure Challenges Faced by Automated Driving: A Review," Applied Sciences, vol. 12, no. 7, p. 3477, 2022.

[8] Á. Arcos-García, J. A. Alvarez-Garcia, and L. M. Soria-Morillo, "Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods," Neural Networks, vol. 99, pp. 158-165, 2018.

[9] J. Zhang, M. Huang, X. Jin, and X. Li, "A real-time Chinese traffic sign detection algorithm based on modified YOLOv2," Algorithms, vol. 10, no. 4, p. 127, 2017.

[10] M. Shahud, J. Bajracharya, P. Praneetpolgrang, and S. Petcharee, "Thai traffic sign detection and recognition using convolutional neural networks," in 2018 22nd International Computer Science and Engineering Conference (ICSEC), 2018: IEEE, pp. 1-5.

[11] D. A. Alghmgham, G. Latif, J. Alghazo, and L. Alzubaidi, "Autonomous traffic sign (ATSR) detection and recognition using deep CNN," Procedia Computer Science, vol. 163, pp. 266-274, 2019.

[12] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," Neural networks, vol. 32, pp. 323-332, 2012.

[13] J. Jin, K. Fu, and C. Zhang, "Traffic sign recognition with hinge loss trained convolutional neural networks," IEEE transactions on intelligent transportation systems, vol. 15, no. 5, pp. 1991-2000, 2014.

[14] Y. Yang, H. Luo, H. Xu, and F. Wu, "Towards real-time traffic sign detection and classification," IEEE Transactions on Intelligent transportation systems, vol. 17, no. 7, pp. 2022-2031, 2015.

[15] F. Jurišić, I. Filković, and Z. Kalafatić, "Multiple-dataset traffic sign classification with OneCNN," in 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 2015: IEEE, pp. 614-618.

[16] J. Li and Z. Wang, "Real-time traffic sign recognition based on efficient CNNs in the wild," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 3, pp. 975-984, 2018.

[17] A. Wong, M. J. Shafiee, and M. S. Jules, "MicronNet: a highly compact deep convolutional neural network architecture for real-time embedded traffic sign classification," IEEE Access, vol. 6, pp. 59803-59810, 2018.

[18] H. Zhang, Z. Cao, Z. Yan, and C. Zhang, "Sill-net: Feature augmentation with separated illumination representation," arXiv preprint arXiv:2102.03539, 2021.

[19] Y. Zhu and W. Q. Yan, "Traffic sign recognition based on deep learning," Multimedia Tools and Applications, vol. 81, no. 13, pp. 17779-17791, 2022.

[20] N. Youssouf, "Traffic sign classification using CNN and detection using faster-RCNN and YOLOV4," Heliyon, vol. 8, no. 12, 2022.

[21] X. R. Lim, C. P. Lee, K. M. Lim, T. S. Ong, A. Alqahtani, and M. Ali, "Recent Advances in Traffic Sign Recognition: Approaches and Datasets," Sensors, vol. 23, no. 10, p. 4674, 2023.

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

[23] R. Timofte, Belgium traffic sign classification (BTSC) dataset, 2014, https://btsd.ethz.ch/shareddata/ (accessed 2023-5-21).

[24] S. Wang, L. Huang, and J. Hu, "Text line detection from rectangle traffic panels of natural scene," in Journal of Physics: Conference Series, 2018, vol. 960, no. 1: IOP Publishing, p. 012038.

[25] A. El-Sawy, H. El-Bakry, and M. Loey, "CNN for handwritten arabic digits recognition based on LeNet-5," in Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016 2, 2017: Springer, pp. 566-575.

[26] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in Proceedings of the 27th international conference on machine learning (ICML-10), 2010, pp. 807-814.

[27] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in Artificial Neural Networks–ICANN 2010: 20th International Conference, Thessaloniki, Greece, September 15-18, 2010, Proceedings, Part III 20, 2010: Springer, pp. 92-101.

[28] Scikit-optimize: sequential model-based optimization in Python — scikit-optimize 0.8.1 documentation. https://scikit-optimize.github.io/stable/index.html (accessed 2023-5-21).

[29] Bayesian optimization with skopt — scikit-optimize 0.8.1 documentation. https://scikit-optimize.github.io/stable/auto_examples/bayesian-optimization.html (accessed 2023-5-21).

[30] Skopt.Space — scikit-optimize 0.8.1 documentation. https://scikit-optimize.github.io/stable/modules/generated/skopt.Space.html (accessed 2023-5-21).

[31] D. Passos and P. Mishra, "A tutorial on automatic hyperparameter tuning of deep spectral modelling for regression and classification tasks," Chemometrics and Intelligent Laboratory Systems, p. 104520, 2022.

[32] M. Karnati, A. Seal, G. Sahu, A. Yazidi, and O. Krejcar, "A novel multi-scale based deep convolutional neural network for detecting COVID-19 from X-rays," Applied Soft Computing, vol. 125, p. 109109, 2022.

[33] Y. Yu, J. Li, C. Wen, H. Guan, H. Luo, and C. Wang, "Bag-of-visual-phrases and hierarchical deep models for traffic sign detection and recognition in mobile laser scanning data," ISPRS journal of photogrammetry and remote sensing, vol. 113, pp. 106-123, 2016.