

Article

Traffic Sign Recognition Based on Bayesian Angular Margin Loss for an Autonomous Vehicle

Taehyeon Kim * , Seho Park and Kyoungtaek Lee

Contents Convergence Research Center, Korea Electronics Technology Institute, Seoul 03924, Republic of Korea

* Correspondence: taehyeon.kim@keti.re.kr

Abstract: Traffic sign recognition is a pivotal technology in the advancement of autonomous vehicles as it is critical for adhering to country- or region-specific traffic regulations. Defined as an image classification problem in computer vision, traffic sign recognition is a technique that determines the class of a given traffic sign from input data processed by a neural network. Although image classification has been considered a relatively manageable task with the advent of neural networks, traffic sign classification presents its own unique set of challenges due to the similar visual features inherent in traffic signs. This can make designing a softmax-based classifier problematic. To address this challenge, this paper presents a novel traffic sign recognition model that employs angular margin loss. This model optimizes the necessary hyperparameters for the angular margin loss via Bayesian optimization, thereby maximizing the effectiveness of the loss and achieving a high level of classification performance. This paper showcases the impressive performance of the proposed method through experimental results on benchmark datasets for traffic sign classification.

Keywords: angular margin loss; autonomous vehicle; Bayesian optimization; computer vision; traffic sign recognition



Citation: Kim, T.; Park, S.; Lee, K. Traffic Sign Recognition Based on Bayesian Angular Margin Loss for an Autonomous Vehicle. *Electronics* **2023**, *12*, 3073. <https://doi.org/10.3390/electronics12143073>

Academic Editor: Yue Wu

Received: 15 June 2023

Revised: 1 July 2023

Accepted: 12 July 2023

Published: 14 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the extraordinary advancement of deep learning technology, a host of computer vision problems have been addressed, with ample alternative solutions being developed and researched. The enhancement of performance by impressive computer vision algorithms has stimulated a plethora of new and emerging industries such as smart cities [1–3], smart factories [4,5], autonomous vehicles [6–10], and the metaverse [11,12], all founded on computer vision technology. Specifically, autonomous vehicles increasingly employ various multisensory fusion technologies, including LiDAR, to simultaneously input and process camera inputs along with other sensor inputs via neural networks. This results in advanced autonomous driving capabilities, a trend being adopted across various industries and in academia [6,8,10].

The application of computer vision technology to autonomous vehicles can be broadly classified into two categories according to the application: dynamic object recognition and static object recognition. The former involves the identification and detection of objects that can move independently, such as people and cars. Conversely, static object recognition pertains to the identification of stationary objects, such as road signs, road markings, and traffic lights. These objects can provide critical information about the rules and regulations of the road on which the vehicle is operating. Traffic sign recognition presents a unique challenge because unlike other objects, it requires the definition of new models based on the regions where autonomous vehicles operate. For instance, traffic sign regulations differ between the United States and South Korea, necessitating distinct models for each. Moreover, most traffic signs have similar visual features. Although each sign is classified as a distinct class based on arrow direction or numbers, a straightforward neural network-based image classifier is inadequate for yielding satisfactory recognition effects. Further, a

traffic sign recognition model for autonomous vehicles must exhibit superior performance to ensure the safety of all stakeholders, including the user, nearby pedestrians, and the drivers of adjacent vehicles. This is imperative as recognition accuracy directly impacts safety. Consequently, this paper proposes a new high-performance traffic sign recognition algorithm that is ready for integration into current and future autonomous vehicles.

Traffic sign recognition was the subject of numerous studies prior to the surge in the popularity of neural networks. Traditional machine learning techniques, including support vector machine-based methods [13–15], principal component analysis-based methods [16–18], and k-nearest neighbor-based methods [19], have been suggested for traffic sign recognition. However, the performance of traffic sign recognition employing conventional machine learning (such as SVM, PCA, and k-NN) severely deteriorates under various environmental changes, such as low light or rain. Therefore, these methods prove challenging to implement in practical autonomous vehicles. Past studies have only applied various machine learning techniques to traffic sign recognition without further investigating more advanced approaches.

Following the emergence of neural networks and the verification of the high performance and effectiveness of computer vision using convolutional neural networks, traffic sign recognition has surpassed its previous limitations [20–23], much like other image recognition fields. Early neural network-based traffic sign recognition methods suggested ways to deduce the traffic sign class of a given input image by processing a cropped image containing only the traffic sign. More recently, object detection has broadened the scope of traffic sign recognition, enabling the extraction of traffic sign positions from various locations in driving images and the concurrent inference of their respective classes [24–26]. Although deep learning-based traffic sign recognition has outperformed conventional machine learning and signal processing-based approaches, most research has been restricted to retraining other computer vision models using traffic sign recognition datasets or merely applying well-known object detection models like You Only Look Once (YOLO) [27–29]. Hence, while the field of traffic sign recognition has achieved a degree of good performance and expansion, most research is confined to utilizing neural network models with high computational and memory complexity or leveraging general image classifier losses such as cross-entropy. We propose a neural network-based classifier specialized for traffic sign recognition that transcends the level of prior research, which primarily relied on existing well-known neural network models.

The proposed traffic sign recognition method comprises two principal components: angular margin loss and Bayesian optimization. Angular margin loss is a classification loss employed in metric learning and is a frequently used learning strategy in fields like few-shot learning [30–33]. The objective of this loss is to optimize the angles between classes in the feature space during the classification process. Specifically, it optimizes the clustering of features belonging to each class around the center of that class while maintaining a margin between classes in the feature space. This fosters clearer differentiation between classes and is expected to enhance performance. Angular margin loss has gained popularity in recent years, especially in areas like few-shot learning. Bayesian optimization is a machine learning-based optimization technique commonly used in hyperparameter optimization [34–37]. It aims to optimize a black box function, which is typically expensive to evaluate, by iteratively selecting the next evaluation point based on the model's posterior distribution of the function. Bayesian optimization offers an efficient and effective means of finding a model's optimal hyperparameters and has been applied in various fields, including computer vision and natural language processing. We propose a new traffic sign recognition model utilizing both angular margin loss and Bayesian optimization to inherit the strengths of both techniques.

In the field of traffic sign recognition, signs are often visually similar, making it challenging to distinguish between them. Consequently, we determined that using a softmax-based classifier alone would only yield a certain performance level. To address this limitation, we opted to employ angular margin loss. This approach allows us to achieve

a high level of classification performance, even for signs with similar morphological features. However, achieving a high level of classification performance with angular margin loss involves adding margin values to the feature vector formation process in a penalty optimization-like manner. Incorrect margin value definition can lead to various performance degradation issues, such as limiting the neural network's learning convergence. To overcome this challenge and select the optimal margin value, the proposed method employs Bayesian optimization. By incorporating Bayesian optimization, we aim to improve traffic sign recognition performance and provide a more specialized and efficient approach in this field. We believe that our proposed method will make a significant contribution to advancing traffic sign recognition and addressing the challenges associated with visually similar signs.

The rest of this paper is organized as follows. Section 2 provides preliminaries, discussing the Bayesian optimization and angular margin loss techniques in detail. Section 3 presents our proposed method, outlining the architecture and training strategy of our traffic sign recognition model. In Section 4, we present experimental results, demonstrating the efficacy of our approach using various benchmark datasets. Finally, in Section 5, we summarize this paper's main contributions and provide directions for future work.

2. Preliminaries

This section introduces the concepts of angular margin loss and Bayesian optimization, which are integral to our proposed method. Angular margin loss, used in metric learning, optimizes the angle between classes in feature space during classification. This results in features of the same class clustering around their respective center, enhancing class separation and maintaining margins between classes, leading to clearer separations and improved performance. Bayesian optimization, a machine learning-based optimization technique, is typically employed for hyperparameter optimization. It constructs a probabilistic model of the objective function and selects the next hyperparameters to evaluate based on their predicted utility. This approach enables the efficient exploration of hyperparameter space, ultimately improving model performance.

2.1. Metric Learning

Metric learning is a machine learning subfield that aims to learn a distance or similarity metric between data points. Its objective is to transform data such that similar points are proximate in the new metric space, while dissimilar points are distant. This transformation often involves mapping from the original feature space to a new metric space, wherein distances reflect the degree of similarity or dissimilarity between points.

Angular margin loss is a classification loss commonly employed in training neural networks for metric learning tasks like face recognition [38]. It facilitates the learning of a discriminative embedding space where similar samples map to close points and dissimilar samples map to distant points. Angular margin loss introduces a margin between class embeddings, enforcing a specific angular distance and therefore enhancing class separation. By minimizing the angular margin loss, the network learns to map similar samples close to each other while placing dissimilar samples far apart in the embedding space. This process results in a more discriminative embedding space, which is useful for various downstream tasks. Therefore, angular margin loss is instrumental in training neural networks to create an embedding space that clearly separates different classes.

2.2. Angular Margin Loss

The softmax function is commonly used in classification problems to produce a probability distribution over K classes. Given a vector z of K -dimensional logits, the softmax

function maps each element of z to a probability value between 0 and 1, such that the sum of the probabilities over all K classes equal 1. The softmax function is defined as follows:

$$\sigma(z)_j = \frac{\exp(z_j)}{\sum_{k=1}^K \exp(z_k)}, \quad (1)$$

where z_j is the j th element of the vector z , and $\sigma(z)_j$ is the j th element of the softmax output.

The angular margin loss is a loss function used in metric learning to train a neural network to learn a discriminative embedding space. It is designed to learn mapping from the original feature space to a new metric space, where the distances between points reflect their similarity or dissimilarity. The angular margin loss is defined as follows:

$$L_i = -\log \left(\frac{\exp(s(y_i) \cdot (\cos(m_1\theta_{y_i}) - m_2))}{\exp(s(y_i) \cdot (\cos(m_1\theta_{y_i}) - m_2)) + \sum_{j=1, j \neq y_i}^K \exp(s(j) \cdot \cos(\theta_j))} \right) \quad (2)$$

where y_i is the true class label of the i th sample, θ_{y_i} is the angle between the embedding of the i th sample and the weight vector of its true class, θ_j is the angle between the embedding of the i th sample and the weight vector of the j th class, m_1 and m_2 are the margins, and $s(\cdot)$ is a scaling factor.

The softmax function's numerator reflects the cosine similarity between the i th sample's embedding and its true class's weight vector, multiplied by a margin-based angular scaling factor. The denominator represents the cumulative cosine similarities between the i th sample's embedding and the weight vectors of all other classes. Taking the softmax function's negative logarithm derives the i th sample's loss, L_i . This loss penalizes the network when the angular distance between a positive sample's embedding and a negative one is smaller than the margin. It encourages the network to learn embeddings with large angular separations between different classes while keeping similar samples close. The angular margin loss is formulated using the softmax function in which the numerator corresponds to the cosine similarity between the embedding of the i th sample and the weight vector of its true class, multiplied by a margin-based angular scaling factor. The denominator corresponds to the sum of the cosine similarities between the embedding of the i th sample and the weight vectors of all other classes.

The hyperparameters m_1 and m_2 regulate the margin and scaling factor in the angular margin loss formula. The margin m_1 controls the minimum angular distance between the embeddings of a positive sample a negative sample. A larger margin increases the angular separation between embeddings of different classes, making the embedding space more discriminative. The scaling factor m_2 adjusts the scale of cosine similarities in the loss function, allowing the network to fine-tune the separation of similar classes while keeping their embeddings close. These hyperparameters play important roles in controlling the discriminative power and scaling of the loss function.

If the margin m_1 is too small, the network may fail to separate different classes, causing overlaps in the embedding space. Conversely, a large margin makes the network too conservative, leading to excessively distant embeddings for similar samples. It is important to set an appropriate margin that balances class separation and the compactness of similar samples. If the scaling factor m_2 is too small, the loss function may focus too much on cosine similarities and not enough on angular separation, resulting in poor discriminative ability. Conversely, a large scaling factor overemphasizes angular separation and may miss opportunities to separate similar classes. Tuning m_1 and m_2 carefully is crucial for optimizing the balance between class separation and the compactness of similar samples, ensuring optimal performance in downstream tasks.

2.3. Bayesian Optimization

Bayesian optimization is a machine learning-based optimization strategy that is particularly effective for optimizing costly, black box, and derivative-free functions. It consists of two main components: a Bayesian statistical model and an acquisition function [34,36]. The

Bayesian statistical model, usually a Gaussian process, models the objective function and measures the uncertainty of the function at unobserved data points. Initially trained on a small set of input–output pairs, this model is updated iteratively with new data. It predicts the objective function values at new input points and estimates the associated uncertainties. The acquisition function guides the selection of the next input point to evaluate, balancing exploration and exploitation by evaluating the predictive improvement at unobserved points. Common acquisition functions include the probability of improvement, expected improvement, and the upper confidence bound.

Bayesian optimization has several advantages over other optimization techniques. It does not require structural information about the objective function, rendering it suitable for black box optimization in which the underlying function is unknown or complex. It can operate without observing the derivatives of the objective function, which is ideal for derivative-free optimization. Furthermore, it can identify the global optimum by estimating uncertainty at unobserved points, making it effective for global optimization problems. Widely applied to various optimization problems, Bayesian optimization has significantly improved the performance of machine learning algorithms, including deep neural networks, via optimal hyperparameter determination [35,37]. The process of hyperparameter optimization using Bayesian optimization begins with defining the hyperparameter search space. An acquisition function is selected to guide the search. This function measures the potential benefit of evaluating each set of hyperparameters and decides the next set to evaluate. Common choices include expected improvement, the probability of improvement, and the upper confidence bound. The Bayesian optimization process iterates. The initial set of hyperparameters is randomly sampled from the search space, and the corresponding loss function is evaluated. Evaluation results update the Bayesian statistical model, which then selects the next set of hyperparameters for evaluation. This continues until a stopping criterion, such as the maximum number of iterations or hyperparameter convergence, is met. The final set of hyperparameters is selected based on the best performance on the validation set.

Overall, employing Bayesian optimization to tune the hyperparameters of the angular margin loss can improve performance across tasks like image classification, object detection, and face recognition. By automating the hyperparameter tuning process, Bayesian optimization can save substantial time and resources, enabling more efficient development and deployment of machine learning models.

3. Proposed Method

This chapter aims to describe a new traffic sign recognition model that combines angular margin loss and Bayesian optimization. By combining these two techniques, the proposed model achieves greater performance and efficiency compared to traditional softmax-based classifiers. The entire process of the proposed method has been summarized in Algorithm 1 as a pseudo-code.

Algorithm 1. Basic Pseudo-Code for Proposed Method**Input:** Validation dataset V , Initial hyperparameters m_1 and m_2 .**Initialize:**Angular margins $m = [m_1, m_2]$ Define objective function $L(m)$ as average classification error on validation datasetInitialize Gaussian process with mean μ and covariance Σ

While stopping criteria is not met:

 For $t = 1$ to T : Compute prior distribution: $L(m_{1:t}) \sim \mathcal{N}(\mu_0(m_{1:t}), \Sigma_0(m_{1:t}, m_{1:t}))$ Calculate the conditional distribution: $L(m)|L(m_{1:t}) \sim \mathcal{N}(\mu_t(m_{1:t}), \sigma_t^2(m))$ Calculate $\mu_t(m)$: $\mu_t(m) = \Sigma_0(m, m_{1:t})\Sigma_0(m, m_{1:t})^{-1}(L(m_{1:t}) - \mu_0(m_{1:t})) + \mu_0(m_{1:t})$, Calculate $\sigma_t^2(m)$: $\sigma_t^2(m) = \Sigma_0(m, m) - \Sigma_0(m, m_{1:t})\Sigma_0(m_{1:t}, m_{1:t})^{-1}\Sigma_0(m_{1:t}, m)$ Define expected improvement function $\mathcal{E}_t(m) = \mathbb{E}[\max(L(m_{best}) - L(m))]$, Compute $\mathcal{E}_t(m)$ for all m in the search space Select m_{next} with highest $\mathcal{E}_t(m)$ Evaluate $L(m_{next})$ to get new observation

Update Gaussian process with new observation

 Set m_{best} to m_{next} if $L(m_{next}) < L(m_{best})$ **End While****Output:** Optimal hyperparameters m_{best} *Bayesian Angular Margin Loss*

The proposed system in this paper utilizes Bayesian optimization to optimize the hyperparameters of the angular margin loss, aiming to maximize its discriminative performance. Specifically, Bayesian optimization is used to optimize the margin m_1 and scaling factor m_2 in the angular margin loss, automatically searching for the best hyperparameter values. By optimizing the hyperparameters through Bayesian optimization, the system achieves better separation between different classes of traffic signs while maintaining compactness for similar signs. Bayesian optimization is particularly suitable for this task as it efficiently explores the high-dimensional hyperparameter space and identifies the best hyperparameters with a limited number of evaluations. This is valuable in traffic sign recognition, in which hyperparameter optimization can be time-consuming and computationally expensive. The combination of Bayesian optimization and angular margin loss is referred to as Bayesian angular margin loss.

The objective function for the proposed method is defined as finding the hyperparameters m_1 and m_2 that result in the lowest classification error on the validation dataset. It involves a vector \mathbf{m} consisting of m_1 and m_2 , and Bayesian optimization is employed to search for the optimal values of \mathbf{m} . Formally, the objective function can be defined as follows:

$$\min_{\mathbf{m}} L(\mathbf{m}) = \min_{\mathbf{m}} \frac{1}{N_{val}} \sum_{i=1}^{N_{val}} L_i(\mathbf{m}). \quad (3)$$

The objective function minimizes the loss value $L(\mathbf{m})$ averaged over the instances in the validation dataset, denoted as $L_i(\mathbf{m})$. By maximizing the discriminative power of the angular margin loss, the proposed method aims to achieve a high level of performance in traffic sign recognition tasks that involve datasets with mixed visual representations.

The Gaussian process provides a way to model the distribution of the objective function $L(\mathbf{m})$ at unobserved margin values \mathbf{m} , based on the distribution obtained from previously observed margin value combinations $\mathbf{m}_{1:t} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_t\}$. By applying Bayesian inference, we can define the prior distribution $L(\mathbf{m}_{1:t})$ and obtain the conditional distribution $L(\mathbf{m})|L(\mathbf{m}_{1:t})$. To define the prior distribution $L(\mathbf{m}_{1:t})$, we assume that it follows a multivariate normal distribution with a mean and covariance that can be learned from the observed data. This allows us to complete the correlation between different hyper-

parameters and make informed predictions about the objective function at unobserved hyperparameters. Therefore, the prior distribution of the proposed objective function can be defined as follows:

$$L(m_{1:t}) \sim \mathcal{N}(\mu_0(m_{1:t}), \Sigma_0(m_{1:t}, m_{1:t})), \quad (4)$$

where $\mu_0(m_{1:t}) \in \mathbb{R}^t$ is the mean vector, and $\Sigma_0(m_{1:t}, m_{1:t}) \in \mathbb{R}^{t \times t}$ denotes the Mahalanobis covariance matrix, both of which are t -dimensional. The multivariate normal distribution function is denoted by $\mathcal{N}(\cdot, \cdot)$. Using the prior distribution, we can calculate the conditional distribution of $L(m)$ given the observed hyperparameters $m_{1:t}$ as follows:

$$L(m)|L(m_{1:t}) \sim \mathcal{N}(\mu_t(m_{1:t}), \sigma_t^2(m)), \quad (5)$$

where $\mu_t(m_{1:t})$ and $\sigma_t^2(m)$ are the mean and variance of the conditional distribution, respectively. The mean $\mu_t(m_{1:t})$ is given by:

$$\mu_t(m) = \Sigma_0(m, m_{1:t})\Sigma_0(m, m_{1:t})^{-1}(L(m_{1:t}) - \mu_0(m_{1:t})) + \mu_0(m_{1:t}), \quad (6)$$

where $\Sigma_0(m, m_{1:t})^{-1}$ denotes the inverse of the Mahalanobis covariance matrix, and $\mu_0(m_{1:t})$ is the mean vector of the prior distribution. The variance $\sigma_t^2(m)$ is given by:

$$\sigma_t^2(m) = \Sigma_0(m, m) - \Sigma_0(m, m_{1:t})\Sigma_0(m_{1:t}, m_{1:t})^{-1}\Sigma_0(m_{1:t}, m) \quad (7)$$

where $\Sigma_0(m, m)$ is the covariance of the prior distribution. By using the conditional distribution, we can estimate the uncertainty of the objective function at unobserved hyperparameters, which allows us to efficiently search the hyperparameter space using Bayesian optimization. After applying a Gaussian process to the observed margin combinations $m_{1:t}$, we can estimate the potential values of the objective function at unobserved margin combination m using a conditional distribution. To determine the next margin combination to evaluate, we use the expected improvement acquisition function.

The expected improvement compares the loss of the current best margin combination, denoted as m_{best} , with the estimated objective error of the candidate margin combinations m . The objective error is defined as the difference between the loss of the candidate margin combination and the loss of the current best margin combination. The expected improvement can be expressed as:

$$\mathcal{E}_t(m) = \mathbb{E}[\max(L(m_{best}) - L(m))], \quad (8)$$

where $\mathbb{E}[\cdot]$ denotes the expected value, $L(m_{best})$ is the loss of the current best margin combination, and $L(m)$ is the estimated loss of the candidate margin combination m based on the conditional distribution. By using the expected improvement acquisition function, we can identify the margin combination that is expected to yield the largest improvement in the objective function and evaluate it in the next iteration. By repeating this process, we can efficiently search the space of margin combinations and find the optimal combination that yields the lowest loss. It should be noted that the proposed objective function computes the loss of the observed margin combination without noise, which allows us to determine the current best margin combination m_{best} as the one that yields the lowest loss among the observed margin combinations.

Based on the process described above, the proposed technique effectively determines optimal margin values for the angular margin loss, resulting in the improved utilization of the loss function for traffic sign recognition. Despite Gaussian processes imposing constraints on hyperparameter dimensionality, the number of margins in the angular margin loss is sufficient for convergence, making Bayesian optimization capable of providing globally optimal margin values.

In summary, this paper introduces a traffic sign recognition system that employs Bayesian optimization to optimize the hyperparameters of the angular margin loss. The primary objective is to maximize the discriminative performance of the system by achieving better separation between different classes of traffic signs while maintaining compactness for similar signs. The combined approach of Bayesian optimization and angular margin loss is referred to as Bayesian angular margin loss. The proposed objective function aims to find the optimal values of m_1 and m_2 in the angular margin loss, leading to the lowest classification error on the validation dataset. By utilizing Gaussian processes and Bayesian inference, the distribution of the objective function at unobserved margin values can be modeled, allowing for the definition of prior and conditional distributions. This technique provides optimal margin values for the angular margin loss, thereby improving its effectiveness for traffic sign recognition.

4. Experimental Results

This chapter covers various experimental results and their analysis for evaluating the performance of the proposed technique for recognizing traffic signs.

4.1. Implementation Details

In this paper, we conducted quantitative evaluations of the proposed technique using a total of two benchmark datasets. The first dataset used was the German Traffic Sign Recognition Benchmark (GTSRB) [39], and the second was the Traffic-Sign Detection and Classification in the Wild (TT100K) dataset [40]. GTSRB is a dataset containing traffic sign images that is widely used for benchmarking traffic sign recognition algorithms. The dataset contains more than 50,000 images of traffic signs from 43 different classes. The images were captured under various lighting and weather conditions, and they include a wide range of traffic sign variations such as occlusion, blur, and different viewing angles. In the experiment in this paper, we constructed a training set containing 8514 images from the GTSRB and a validation set containing 2157 images. TT100K is a large-scale dataset of traffic sign images that is designed to evaluate traffic sign detection and classification algorithms under challenging real-world conditions. The dataset contains more than 100,000 images of traffic signs captured from various locations and in various weather and lighting conditions. The images include a wide range of traffic sign variations, such as occlusion, blur, and deformation, and they cover a wide range of traffic sign types and shapes. The training set has 6107 images, and the validation set has 3073 images. The samples from the GSRB and TT100K dataset are shown in Figure 1.

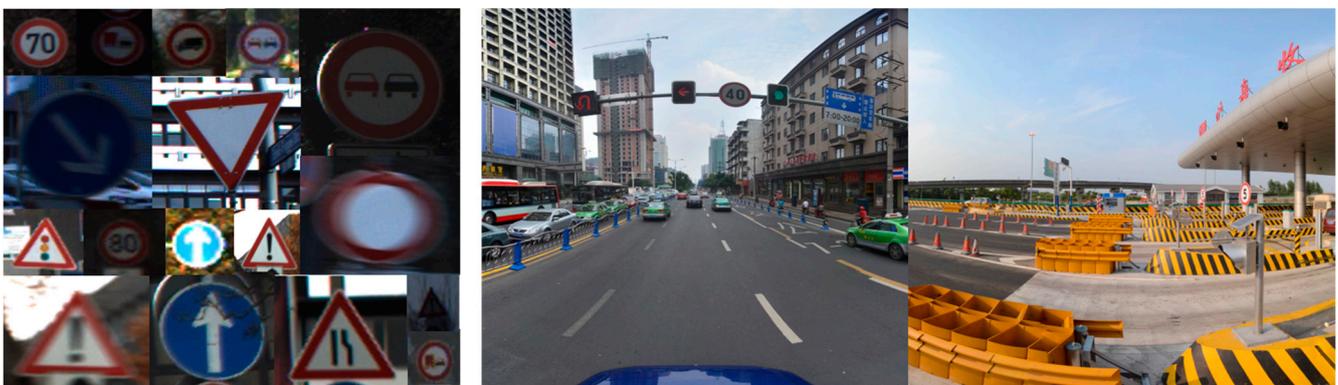


Figure 1. The samples from the GTSRB and TT100K datasets are composed of (left) images from the GTSRB, which include a variety of conditions, such as different lighting environments, camera shakes, and occlusions like those from tree coverage, and (right) samples from the TT100K dataset, which is designed to detect traffic signs from images captured in diverse road conditions. The TT100K dataset is more varied in comparison to the GTSRB, even containing images taken from the sides of traffic signs.

The implementation of the proposed method leveraged the BoTorch framework [41] for the Bayesian optimization algorithm, and the PyTorch framework [42] was used for the training and inference of the neural network. In experiments using the TT100K dataset, not only the performance indicators but also the memory complexity of the model and inference time had to be compared with other recent techniques; hence, an RTX 3080 GPU device was used. The number of iterations in the Bayesian optimization was set to a total of 50, and the ranges of the values m_1 and m_2 , which must be obtained in Bayesian optimization, were set from 0.5 to 1.0 and from 0.1 to 0.5, respectively. This is because if each margin is smaller or larger than the range, performance degradation occurs [30–32]. By controlling the search area of the Bayesian optimization to an appropriate size, we can induce rapid convergence.

4.2. Experimental Results on the GTSRB

Due to its small size and the well-curated images in the training dataset, the GTSRB has already achieved high classification performance using neural networks. Therefore, in this paper, we aimed to visualize the logits of the neural network rather than conduct a quantitative performance evaluation on GTSRB to visually analyze the performance of the proposed technique. We present the visualization of logits from each neural network trained on the GTSRB via a principal component analysis in Figure 2. This experiment utilized the ResNet-18 model, serving as an ablation study on Bayesian optimization. The proposed technique incorporates Bayesian optimization to enhance the discriminative performance of the angular margin loss, an effect that is readily observable in the figure. The sub-figure on the left depicts results using only the angular margin loss. In contrast, the right sub-figure demonstrates the results of the neural network trained using the proposed technique. The comparison between the two figures clearly reveals that our method achieves superior intraclass compactness and interclass discrepancy compared to the traditional approach.



Figure 2. Visualizing high-dimensional logits via a principal component analysis: (left) the logits from the proposed traffic sign recognition without Bayesian optimization (test accuracy: 96.74%); (right) the logits from the proposed traffic sign recognition method (test accuracy: 97.83%).

These findings emphasize the crucial role of appropriate margin settings in angular margin loss-based traffic sign recognition. Furthermore, they affirm that the proposed method can significantly improve traffic sign recognition performance. The technique's effectiveness in enhancing the discriminative capability of the angular margin loss in traffic

sign recognition is clearly demonstrated through the experimental results on the GTSRB. In summary, our technique has great potential to substantially boost the performance of traffic sign recognition systems. It could serve as a valuable tool for a plethora of real-world applications (Table 1).

Table 1. Detection performance of different methods on the TT100K.

Method	Input Size	Backbone	FPS	MAP@.5
YOLOv5s [27]	640 × 640	CSPDarknet	136	80.1
YOLOv7-tiny [28]	640 × 640	E-ELAN	142	84.3
YOLOX [43]	640 × 640	CSPDarknet	55	84.9
AIE-YOLO [44]	640 × 640	CSPDarknet	87	83.5
TRD-YOLO [45]	640 × 640	CSPDarknet	73	86.5
Ours (YOLOv5s)	640 × 640	CSPDarknet	130	83.2
Ours (YOLOv7-tiny)	640 × 640	E-ELAN	138	87.9

4.3. Experimental Results on TT100K

The TT100K dataset, as illustrated in Figure 1, comprises numerous diverse traffic signs within a single image. Thus, it is better to consider this dataset a detection benchmark that evaluates both classification and localization instead of merely a classification benchmark. The primary aim of the experiment was to ascertain the performance enhancement achieved by our proposed technique, which improves sign recognition, when applied to YOLOv5s and YOLOv7-tiny models. For the performance evaluation, we used two crucial metrics: frames per second (FPS) and mean average precision at an intersection over union threshold of 0.5 (MAP@.5).

FPS indicates how many frames the model can process within a second. A higher FPS suggests a faster model capable of handling more data in less time. MAP@.5, on the other hand, is an evaluation metric for object detection models that represents the average precision of the model at various recall levels when the intersection over union (IoU) is 0.5 or above. A higher MAP@.5 suggests a more accurate model. All experiments were carried out on an RTX 3060 GPU with a configured input size of 640 × 640 for all the models.

The baseline YOLOv5s model, leveraging the CSPDarknet backbone, achieved an FPS value of 136 and a MAP@.5 score of 80.1. YOLOv7-tiny, using E-ELAN as its backbone, outperformed YOLOv5s, reaching an FPS value of 142 and a MAP@.5 score of 84.3. Other models such as YOLOX and AIE-YOLO, also employing the CSPDarknet backbone, achieved FPS values of 55 and 87, respectively, with corresponding MAP@.5 scores of 84.9 and 83.5. Notably, TRD-YOLO achieved the highest accuracy, with a MAP@.5 score of 86.5 but at a slightly slower FPS of 73. Upon integrating our proposed method with the YOLOv5s and YOLOv7-tiny models, we observed substantial improvements. The enhanced YOLOv5s model, referred to as “Ours (YOLOv5s),” achieved an FPS value of 130 and a MAP@.5 score of 83.2, indicating improved accuracy despite a slight decrease in speed. Our modified version of the YOLOv7-tiny model, denoted as “Ours (YOLOv7-tiny),” achieved the highest performance with an FPS value of 138 and an outstanding MAP@.5 score of 87.9. As depicted in Figure 3, our proposed method improved the confidence scores of the models in correctly identifying and drawing the bounding box around the signboard class. This enhancement led to an increase in accuracy, especially for the YOLOv7-tiny model, without significantly affecting the processing speed. The balance between speed and accuracy is crucial for object detection models’ real-world applications, proving that our method is effective at achieving this equilibrium.



Figure 3. The TT100K test dataset sample provides the results for traffic sign detection as follows: the first set of results represent detection by the YOLOv5s model; the second set showcases the detection performance of YOLOv5s enhanced with our proposed technique; the third set of results were produced by the tiny-YOLOv7 model; and the fourth set represents the output of the tiny-YOLOv7 model when our proposed technique was applied. All these images are presented in vectorized versions, allowing you to zoom in for a more detailed examination of the experimental results.

5. Conclusions

This paper introduces a novel approach to traffic sign recognition by integrating angular margin loss and Bayesian optimization. The angular margin loss is central to addressing the challenge of morphological similarity among traffic sign classes. Through experimental validation, we established that this approach outperforms the traditional softmax-based classifier. Additionally, we used Bayesian optimization to fine-tune the margin value, a hyperparameter in angular margin loss. This key technological implementation significantly enhances the discriminative power of the proposed model. Our method demonstrated the potential to boost intra-class compactness and overall performance by replacing the softmax loss in the prevailing traffic sign detection models. Future work will focus on enhancing traffic sign detection by merging object detection with traffic sign classification. We also plan to apply a more efficient Bayesian optimization strategy and extend our proposed Bayesian angular margin loss to other computer vision models used in autonomous vehicles, opening up new avenues for improved performance.

Author Contributions: Conceptualization, T.K.; methodology, T.K.; software, T.K.; validation, T.K. and S.P.; formal analysis, T.K.; investigation, T.K.; resources, T.K.; data curation, T.K.; writing—original draft preparation, T.K.; visualization, T.K.; supervision, T.K.; project administration, T.K. and K.L.; funding acquisition, S.P. and K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by the Ministry of Trade, Industry and Energy (MOTIE) and Korea Institute of Advancement of Technology (KIAT) through the International Cooperative R&D Program (P0019782, Embedded AI Based Fully Autonomous Driving Software and MaaS Technology Development).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Okai, E.; Feng, X.; Sant, P. Smart cities survey. In Proceedings of the 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Exeter, UK, 28–30 June 2018.
2. Pellicer, S.; Santa, G.; Bleda, A.L.; Maestre, R.; Jara, A.J.; Skarmeta, A.G. A global perspective of smart cities: A survey. In Proceedings of the 2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Taichung, Taiwan, 3–5 July 2013.
3. Syed, A.S.; Sierra-Sosa, D.; Kumar, A.; Elmaghraby, A. IoT in smart cities: A survey of technologies, practices and challenges. *Smart Cities* **2021**, *4*, 429–475. [[CrossRef](#)]

4. Shrouf, F.; Ordieres, J.; Miragliotta, G. Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm. In Proceedings of the 2014 IEEE International Conference on Industrial Engineering and Engineering Management, Selangor, Malaysia, 9–12 December 2014.
5. Kalsoom, T.; Ramzan, N.; Ahmed, S.; Ur-Rehman, M. Advances in sensor technologies in the era of smart factory and industry 4.0. *Sensors* **2020**, *20*, 6783. [[CrossRef](#)] [[PubMed](#)]
6. Ahangar, M.N.; Ahmed, Q.Z.; Khan, F.A.; Hafeez, M. A survey of autonomous vehicles: Enabling communication technologies and challenges. *Sensors* **2021**, *21*, 706. [[CrossRef](#)] [[PubMed](#)]
7. Rasouli, A.; Tsotsos, J.K. Autonomous vehicles that interact with pedestrians: A survey of theory and practice. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 900–918. [[CrossRef](#)]
8. Janai, J.; Güneş, F.; Behl, A.; Geiger, A. Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Found. Trends® Comput. Graph. Vis.* **2020**, *12*, 1–308. [[CrossRef](#)]
9. Eggers, F.; Eggers, F. Drivers of autonomous vehicles—Analyzing consumer preferences for self-driving car brand extensions. *Mark. Lett.* **2022**, *33*, 89–112. [[CrossRef](#)]
10. Thomas, E.; McCrudden, C.; Wharton, Z.; Behera, A. Perception of autonomous vehicles by the modern society: A survey. *IET Intell. Transp. Syst.* **2020**, *14*, 1228–1239. [[CrossRef](#)]
11. Wang, Y.; Su, Z.; Zhang, N.; Xing, R.; Liu, D.; Luan, T.H.; Shen, X. A survey on metaverse: Fundamentals, security, and privacy. *IEEE Commun. Survo. Tutor.* **2022**, *25*, 319–352. [[CrossRef](#)]
12. Jeon, H.J.; Youn, H.C.; Ko, S.M.; Kim, T.H. Blockchain and AI Meet in the Metaverse. In *Advances in the Convergence of Blockchain and Artificial Intelligence*; IntechOpen: London, UK, 2022; p. 73.10.5772.
13. Zaklouta, F.; Stanciulescu, B. Real-time traffic-sign recognition using tree classifiers. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 1507–1514. [[CrossRef](#)]
14. Yao, C.; Wu, F.; Chen, H.J.; Hao, X.L.; Shen, Y. Traffic sign recognition using HOG-SVM and grid search. In Proceedings of the 2014 12th International Conference on Signal Processing (ICSP), Hangzhou, China, 19–23 October 2014.
15. Gomez-Moreno, H.; Maldonado-Bascon, S.; Gil-Jimenez, P.; Lafuente-Arroyo, S. Goal evaluation of segmentation algorithms for traffic sign recognition. *IEEE Trans. Intell. Transp. Syst.* **2010**, *11*, 917–930. [[CrossRef](#)]
16. Fleyeh, H.; Davami, E. Eigen-based traffic sign recognition. *IET Intell. Transp. Syst.* **2011**, *5*, 190–196. [[CrossRef](#)]
17. Ruta, A.; Li, Y.; Liu, X. Real-time traffic sign recognition from video by class-specific discriminative features. *Pattern Recognit.* **2010**, *43*, 416–430. [[CrossRef](#)]
18. Perez-Perez, S.E.; Gonzalez-Reyna, S.E.; Ledesma-Orozco, S.E.; Avina-Cervantes, J.G. Principal component analysis for speed limit Traffic Sign Recognition. In Proceedings of the 2013 IEEE International Autumn Meeting on Power Electronics and Computing (ROPEC), Morelia, Mexico, 13–15 November 2013.
19. Han, Y.; Virupakshappa, K.; Oruklu, E. Robust traffic sign recognition with feature extraction and k-NN classification methods. In Proceedings of the 2015 IEEE International Conference on Electro/Information Technology (EIT), Dekalb, IL, USA, 21–23 May 2015.
20. Luo, H.; Yang, Y.; Tong, B.; Wu, F.; Fan, B. Traffic sign recognition using a multi-task convolutional neural network. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 1100–1111. [[CrossRef](#)]
21. Arcos-García, Á.; Alvarez-García, J.A.; Soria-Morillo, L.M. Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods. *Neural Netw.* **2018**, *99*, 158–165. [[CrossRef](#)]
22. Qian, R.; Yue, Y.; Coenen, F.; Zhang, B. Traffic sign recognition with convolutional neural network based on max pooling positions. In Proceedings of the 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Changsha, China, 13–15 August 2016; IEEE: Piscataway, NJ, USA, 2016.
23. Wu, Y.; Liu, Y.; Li, J.; Liu, H.; Hu, X. Traffic sign detection based on convolutional neural networks. In Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 4–9 August 2013.
24. Song, W.; Suandi, S.A. TSR-YOLO: A Chinese Traffic Sign Recognition Algorithm for Intelligent Vehicles in Complex Scenes. *Sensors* **2023**, *23*, 749. [[CrossRef](#)]
25. Mangshor NN, A.; Paudzi NP, A.M.; Ibrahim, S.; Sabri, N. A Real-Time Malaysian Traffic Sign Recognition Using YOLO Algorithm. In *Proceedings of the 12th National Technical Seminar on Unmanned System Technology 2020: NUSYS'20*; Springer: Singapore, 2022.
26. Dewi, C.; Chen, R.-C.; Jiang, X.; Yu, H. Deep convolutional neural network for enhancing traffic sign recognition developed on Yolo V4. *Multimed. Tools Appl.* **2022**, *81*, 37821–37845. [[CrossRef](#)]
27. Ultralytics/YOLOV5. Available online: <https://github.com/ultralytics/yolov5> (accessed on 14 December 2022).
28. Wang, C.Y.; Bochkovskiy, A.; Liao HY, M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
29. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
30. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019.
31. Kim, T.; Hong, E.; Choe, Y. Deep Morphological Anomaly Detection Based on Angular Margin Loss. *Appl. Sci.* **2021**, *11*, 6545. [[CrossRef](#)]

32. Jiao, J.; Liu, W.; Mo, Y.; Jiao, J.; Deng, Z.; Chen, X. Dyn-arcface: Dynamic additive angular margin loss for deep face recognition. *Multimed. Tools Appl.* **2021**, *80*, 25741–25756. [[CrossRef](#)]
33. Choi, H.; Som, A.; Turaga, P. AMC-loss: Angular margin contrastive loss for improved explainability in image classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020.
34. Frazier, P.I. Bayesian optimization. In *Recent Advances in Optimization and Modeling of Contemporary Problems*; Informs: Catonsville, MD, USA, 2018; pp. 255–278.
35. Kim, T.; Lee, J.; Choe, Y. Bayesian optimization-based global optimal rank selection for compression of convolutional neural networks. *IEEE Access* **2020**, *8*, 17605–17618. [[CrossRef](#)]
36. Snoek, J.; Rippel, O.; Swersky, K.; Kiros, R.; Satish, N.; Sundaram, N.; Patwary, M.; Prabhat, M.; Adams, R. Scalable bayesian optimization using deep neural networks. In Proceedings of the 32nd International Conference on Machine Learning, PMLR, Lille, France, 6–11 July 2015.
37. Kim, T.; Choi, H.; Choe, Y. Automated Filter Pruning Based on High-Dimensional Bayesian Optimization. *IEEE Access* **2022**, *10*, 22547–22555. [[CrossRef](#)]
38. Wang, F.; Cheng, J.; Liu, W.; Liu, H. Additive margin softmax for face verification. *IEEE Signal Process. Lett.* **2018**, *25*, 926–930. [[CrossRef](#)]
39. Stallkamp, J.; Schlipsing, M.; Salmen, J.; Igel, C. The German traffic sign recognition benchmark: A multi-class classification competition. In Proceedings of the 2011 International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August 2011.
40. Zhu, Z.; Liang, D.; Zhang, S.; Huang, X.; Li, B.; Hu, S. Traffic-sign detection and classification in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
41. Balandat, M.; Karrer, B.; Jiang, D.R.; Daulton, S.; Letham, B.; Wilson, A.G.; Bakshy, E. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21524–21538.
42. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
43. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding Yolo Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
44. Yan, B.; Li, J.; Yang, Z.; Zhang, X.; Hao, X. AIE-YOLO: Auxiliary Information Enhanced YOLO for Small Object Detection. *Sensors* **2022**, *22*, 8221. [[CrossRef](#)]
45. Chu, J.; Zhang, C.; Yan, M.; Zhang, H.; Ge, T. TRD-YOLO: A real-time, high-performance small traffic sign detection algorithm. *Sensors* **2023**, *23*, 3871. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.