Self-supervised few-shot learning for real-time traffic sign classification

Anh-Khoa Tho Nguyen [a,1], Tin Tran [b,2], Phuc Hong Nguyen [c,3], Vinh Quang Dinh [a,4,*]

[a] Department of Computer Science, Vietnamese German University, Binh Duong, Vietnam
[b] AI Graduate School, Gwangju Institute of Science and Technology, Gwangju 61005, Republic of Korea
[c] Department of Software Engineering, Eastern International University, Binh Duong, Vietnam
[1] 30421001@student.vgu.edu.vn; [2] ttrungtin@gist.ac.kr; [3] phuc.nguyenhong@eiu.edu.vn; [4] vinh.dq2@vgu.edu.vn
* corresponding author

ARTICLE INFO

Article history
Received July 14, 2023
Revised September 21, 2023
Accepted August 20, 2023
Available online February 29, 2024

Selected paper from The 2023 6th International Symposium on Advanced Intelligent Informatics (SAIN'23), Yogyakarta (Virtually), September 21, 2023, http://sain.ijain.org/2023/. Peer-reviewed by SAIN'23 Scientific Committee and Editorial Team of IJAIN journal.

Keywords
Traffic sign classification
One-shot learning
Few-shot learning
Self-supervised learning
CLIP-based approach

ABSTRACT

Although supervised approaches for traffic sign classification have demonstrated excellent performance, they are limited to classifying several traffic signs defined in the training dataset. This prevents them from being applied to different domains, i.e., different countries. Herein, we propose a self-supervised approach for few-shot learning-based traffic sign classification. A center-awareness similarity network is designed for the traffic sign problem and trained using an optical flow dataset. Unlike existing supervised traffic sign classification methods, the proposed method does not depend on traffic sign categories defined by the training dataset. It applies to any traffic signs from different countries. We construct a Korean traffic sign classification (KTSC) dataset, including 6000 traffic sign samples and 59 categories. We evaluate the proposed method with baseline methods using the KTSC, German traffic sign, and Belgian traffic sign classification datasets. Experimental results show that the proposed method extends the ability of existing supervised methods and can classify any traffic sign, regardless of region/country dependence. Furthermore, the proposed approach significantly outperforms baseline methods for patch similarity. This approach provides a flexible and robust solution for classifying traffic signs, allowing for accurate categorization of every traffic sign, regardless of regional or national differences.

1. Introduction

In the current era of rapid technology development, autonomous driving is a pivotal element that has significantly affected industry and academia. In addition, efficient autonomous systems and human-computer interactions are indispensable. Improving the safety of pedestrians is crucial in the context of autonomous driving. Hence, robots and humans must be on a common platform to understand the environment more effectively. Traffic sign recognition and classification provide a link for human-computer interactions in both the driver assistance system and autonomous driving.

Traffic sign recognition systems are vital for ensuring the safety and efficacy of traffic flow. However, this task is associated with challenges such as illumination variation, partial occlusion, different viewpoints, and weather conditions, rendering it difficult for the computer to detect and recognize traffic signs. Numerous benchmarks have been proposed to mitigate these challenges [1], [2]. Traffic sign recognition can be classified into two modules: detection and classification. During detection, the location of the sign in the images is assimilated, whereas during classification, the sign of the system is determined. Although both modules are independent, they are dependent on each other in a full system.

The breakthrough of deep convolutional neural networks (CNNs) has yielded state-of-the-art results for traffic sign recognition. However, classical deep learning algorithms require a large amount of data and in some cases, the acquisition of sufficient data for performance improvement is unrealistic or challenging. In traffic sign recognition and detection for example, supervised approaches require training data, and collecting a large number of samples is not an easy task. In addition, we cannot use traffic sign dataset from one country and apply to another country as their distribution are different (having different type of signs). In these scenarios, few-shot learning algorithms are utilized to determine the data patterns.

Herein, we proposed a few-shot learning-based traffic classification network that comprises a similarity network based on a CNN. Image patches are generated using standard geometric transformations. To train the similarity network, these input image patches are further subdivided into five smaller patches for feature extraction by overlapping the central patch with the other patches. The primary contribution of this study is the use of self-supervised few-shot learning for traffic sign classification; therefore, the proposed method does not depend on the classification of training data. It classifies the traffic sign provided that examples are available. The proposed method applies to any traffic sign dataset regardless of country, in contrast to supervised methods of classification, which depend significantly on the training dataset and can classify the traffic sign therein explicitly. This training data dependency problem is addressed in this study using a self-supervised method for traffic classification. The proposed method was tested on the German traffic sign (GTS) and Belgian traffic sign classification (BTSC) datasets, and the experimental results show the efficacy of the proposed methodology.

The main contributions of this paper are as follows:

- A novel self-supervised approach for traffic sign classification that does not require a traffic sign dataset in the training step is proposed.
- A center-awareness similarity network is proposed for traffic sign classification. In addition, a simple data augmentation technique is introduced to force networks to focus on the central part of an input image.
- A Korean traffic sign classification dataset is introduced and available online.
- Experimental results using common traffic sign classification datasets show that the proposed method exhibits better generalization than supervised methods and significantly outperforms testing baseline methods.

The remainder of this paper is organized as follows: In Section 2, we present the related work. The proposed methodology and our Korean traffic sign classification dataset are described in Section 3. Experimental results for traffic sign classification are presented in Section 4. Finally, conclusions are provided in Section 5.

2. Related Works

The advent of deep learning has revolutionized traffic sign detection and classification. Zhu *et al.* [3] proposed a deep convolution-based network to detect traffic signs. It uses region proposals to reduce the effective area of search. Moreover, they extended R-CNN for traffic signs and obtained state-of-the-art results. Sermanet *et al.* [4] proposed a convolution neural network with a multi-scale. They modified the architecture by introducing the non-linearity of rectified sigmoid followed by subtractive local normalization and divisive local normalization. Besides commonly used supervised approaches for traffic sign recognition and detection, several other approaches can be used to solve the problem.

2.1. Out-of-distribution

The challenge of distribution shifts in deep learning models, particularly when discrepancies exist between training and test data distributions, is addressed in Zhang *et al.* [5]. A technique is introduced that determines weights for training samples, thereby decoupling features, reducing misleading associations, and placing emphasis on the genuine association between distinguishing attributes and their respective labels. Krueger *et al.* [6] focuses on improving model responses to extreme shifts, particularly when inputs encompass both anti-causal and causal factors.

The Open Domain Generalization (OpenDG) problem, which focuses on the effective training of models across multiple source domains with diverse label sets for optimal performance on unfamiliar target domains, is presented in Shu *et al.* [7]. A framework named Domain-Augmented Meta-Learning (DAML) is introduced, wherein domains are enhanced at the feature level by employing an innovative Dirichlet mixup technique and at the label level through the application of distilled soft labels.

The concept of multi-source open-set unsupervised domain adaptation (MS-OSDA) [8] builds upon the constraints of its single-source counterpart (SS-OSDA). An approach driven by adversarial learning is suggested by the authors, which establishes a communal feature environment across all domains, diminishing discrepancies between multiple origin domains.

Mancini *et al.* [9] address challenges associated with Zero-Shot Learning (ZSL) and Domain Generalization (DG) by the introduction of a unique ZSL+DG scenario, in which the identification of unfamiliar visual concepts in unknown domains is the objective. CuMix, a pioneering method, was introduced by the researchers, and it is designed to emulate domain and semantic variations at test-time by blending visuals and attributes from different foundational domains and classes while being trained. Models designed to generalize to both unfamiliar classes (termed as zero-shot learning) and unseen domains (known as domain generalization) are the focus of Mangla *et al.* [10], and this concept is referred to as zero-shot domain generalization.

2.2. Domain Adaptation

Lu *et al.* [11] proposed the advanced unsupervised domain adaptation (UDA) by utilizing an increased number of classifiers without complicating the model. Through the newly proposed technique, classifiers are depicted using a Gaussian distribution, which permits the generation of a diverse set of classifiers while keeping the model's dimensions comparable to that of two classifiers. Liang *et al.* [12] focused on improving domain adaptation by rectifying biases in classifiers during the knowledge transition from domains abundant in labels to those lacking them. In response, the introduced Auxiliary Target Domain-Oriented Classifier (ATDOC) deploys a specialized classifier tailored for the target domain, enhancing the accuracy of pseudo-labels.

Xu *et al.* [13] presented a fresh approach to Universal Domain Adaptation (UniDA) specifically for remote sensing image scene classification, removing the traditional boundaries between source and target domain label sets. Zhu *et al.* [14] seek to overcome the challenges posed by conventional deep-domain adaptation techniques by emphasizing Subdomain Adaptation, which delves into detailed nuances within categories spanning various domains. In pursuit of this, the Deep Subdomain Adaptation Network (DSAN) is unveiled, leveraging a local maximum mean discrepancy (LMMD) to synchronize pertinent subdomain distributions, eliminating the necessity for adversarial training.

Hu *et al.* [15] present methods of Unsupervised Domain Adaptation (UDA) for identifying and categorizing lanes in self-driving vehicles, utilizing artificial data from virtual settings. Rectifying the mode collapse challenge in adversarial learning methods for unsupervised domain adaptation (UDA) was proposed by Chen *et al.* [16]. In their approach, the researchers propose a unique discriminator-free adversarial learning network (DALN), in which a category classifier takes on the role of a discriminator, guaranteeing alignment across domains and clear category differentiation.

2.3. One- and few-shot learning

Hu *et al.* [17] refine few-shot learning through a straightforward pre-train + ProtoNet method, highlighting the significance of external datasets and neural network design. When domain variations occur, adjusting with data augmentation becomes essential. Zhang *et al.* [18] proposed a Meta-DETR which is an object detection framework that uses DETR's transformer architecture for few-shot learning.

It processes images directly without requiring region proposals, addressing inaccuracies in traditional methods. Lu *et al.* [19] propose an unsupervised few-shot learning method grounded in information theory, using self-supervision to optimize mutual information between data instances and their representations. They emphasize the distinct mutual information (MI) objectives of self-supervised versus supervised pre-training, exploring these differences through thorough experimentation.

Lu *et al.* [20] propose the Contour Transformer Network (CTN), a one-shot segmentation approach that utilizes one labeled reference and several unlabeled images for training. By capitalizing on the uniformity of anatomical structures' shape and visual patterns across images, CTN adopts a semi-supervised method, focusing on contour progression. In medical image segmentation, the effectiveness of supervised neural models is constrained by the demand for a vast number of labeled samples, positioning one-shot learning as a viable solution when annotations are scarce [21]. Drawing inspiration from atlas-centric segmentation, the author presents an innovative self-supervised learning technique that creates volumetric image-segmentation pairs using just one labeled reference. Yang *et al.* [22] highlight the crucial importance of aligning features at the instance level for the progression of one-shot object detection techniques. A distinct IHR (Instance-level Hierarchical Relation) module is unveiled to encapsulate relationships across different levels, refining the depiction of similarities.

2.4. CLIP-based Approach

Radford *et al.* [23] introduce a new method to address the constraints of conventional computer vision systems that are confined to trained classes. By integrating NLP with vision, the model learns from text-image pairs, utilizing a vast dataset of 400 million entries. This technique enables zero-shot transfer across tasks, rivaling ResNet-50's performance on ImageNet without needing its comprehensive training set. Gu *et al.* [24] refine the process of open-vocabulary object detection, allowing for object detection through diverse textual inputs. The introduced technique, ViLD, taps into Vision and Language knowledge Distillation, drawing from an established image classification framework. Using this foundational "teacher" model, it encodes various category narratives and visual areas, subsequently integrating them into a secondary "student" detection system for enhanced performance.

Hendriksen *et al.* [25] target the issue of aligning product categories with their corresponding images in e-commerce due to frequent discrepancies between text and visuals. This approach employs four specialized encoders (for category, image, title, and attributes) and a pair of projection mechanisms to map both category and product details into a cohesive multimodal domain for optimized retrieval. Jiang and Ye [26] focuse on enhancing the process of Image Person Retrieval based on text descriptions through the IRRA strategy, utilizing techniques inspired by the CLIP model. This strategy incorporates a unique application of Masked Language Modeling to subtly identify connections between visual and textual data.

Ge *et al.* [27] amplify the resilience and versatility of multi-modal frameworks, with a focus on rectifying the precision variance observed in CLIP's ImageNet zero-shot assessments. By manipulating images and textual indicators, the certainty of predictions is gauged, highlighting predictions that might be off-mark. A hierarchical framework from WordNet is then harnessed to propose a label augmentation technique, utilizing both broad and specific category information to enhance the alignment between visual and textual signals.

Sanghi *et al.* [28] present a technique to create 3D forms using text prompts, tackling the issue of scarcely matched text and shape datasets. CLIP-Forge utilizes a dual-phase training approach: initially, it trains an autoencoder to establish a latent representation for shapes, and subsequently, it incorporates a normalizing flow model that relies on features from a previously trained image encoder. In the inference phase, the system draws on textual attributes from an established text encoder to produce the corresponding 3D form.

3. Method

3.1. Preparation and Patch Transformation

In this subsection, we present an approach for constructing a dataset using an optical flow method, which is then used to train a similarity network. For a specific video, we extracted two frames and used the optical flow method [29] to compute the corresponding points between the frames, as shown in Fig. 1. For each pair of corresponding points, we extracted image patches whose center pixels are the corresponding points. Traffic sign images in the same categories also have similar appearances except for their backgrounds. Fig 1(a) and (b) motivate us to design a self-supervised similarity network for traffic sign classification that focuses on the center part of input images

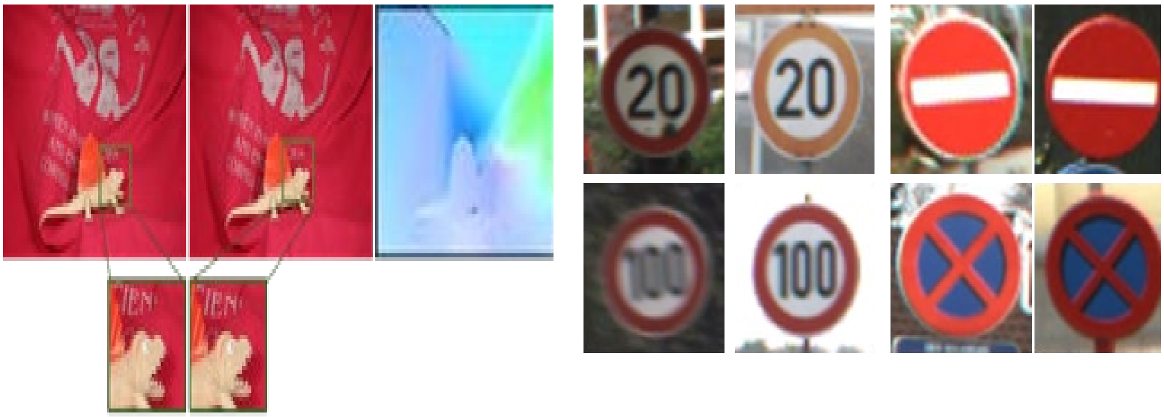
(a) (b)

Fig. 1. Corresponding patches are extracted using optical flow images and sample traffic sign images in different categories. (a) Pairs of two patches are cropped that are visually similar. (b) Traffic signs in different categories.

According to Meister [30], challenges in traffic sign classification include occlusion, illumination variations, snow, sun, rain, and blur. Therefore, the extracted patches were processed to address the challenges. Each patch underwent a pipeline of typical image transformations, such as rotation, translation, scale, elastic distortion, noise addition, and brightness and contrast changes. The brightness and contrast adjustment step changes the brightness and contrast by setting the image patch P as

$$\mathrm{P} \leftarrow \mathrm{P.constrat} + \mathrm{brightness} \tag{1}$$

where addition and multiplication are element-wise operations. The rotation step rotates the patch by rotation, whereas the translation step translates the patch in the vertical direction by translation. The scaling step resizes the patch by scaling, and the shearing step shears the patch in the horizontal direction by shearing. Elastic distortion [31] is typically used to generate feasible and label-preserving images for classification. Elastic distortion changes an image patch by the transformation intensity EDalpha and transformation smoothness EDsigma.

Finally, Gaussian noise was added to each generated patch. We used a Gaussian function to compute the probability that a pixel should be added noise. Let HP and WP be the height and width of a patch P, and c is the center pixel of P where $x_c = \frac{W_p}{2} + 1$ and $y_c = \frac{W_p}{2} + 1$. The probability $prob(x, y)$ that a pixel should be added with noise is computed as

$$\mathrm{prob(x, y)} = 1 - \mathrm{G(x, y)}, \tag{2}$$

where

$$G(x, y) = \frac{1}{2\pi\sigma^2} e - \frac{(x-x_c)+(y-y_c)}{2\sigma^2} \tag{3}$$

To prepare training data for positive and negative examples, pairs of corresponding patches were extracted and subjected to the transformation pipeline with different random parameter settings. The pairs of transformed patches formed positive examples. Negative examples were created using extracted image patches far from the corresponding image patches at a distance data distance.

3.2. Center-awareness Similarity Network

Herein, we propose a similarity network that is biased to exploit information from the center of an image patch. The input patch was divided into five smaller patches which extracted the features independently. Among these smaller patches, one patch overlapped with other patches and was extracted from the center region of the original patch. This is because the center region contained more information to classify traffic signs.

Fig. 2 shows the architecture of the proposed center-awareness similarity network. The architecture of the five weight-sharing sub-networks comprised several convolution layers, followed by a rectified linear unit (ReLU) layer, and a max pooling layer. The resulting ten vectors were concatenated and forwardly propagated through a series of fully connected layers, followed by the ReLU layer. The final output of the network was fed to a nonlinear activation function sigmoid to produce a similarity score between the input patches. The binary cross-entropy loss was used for training.

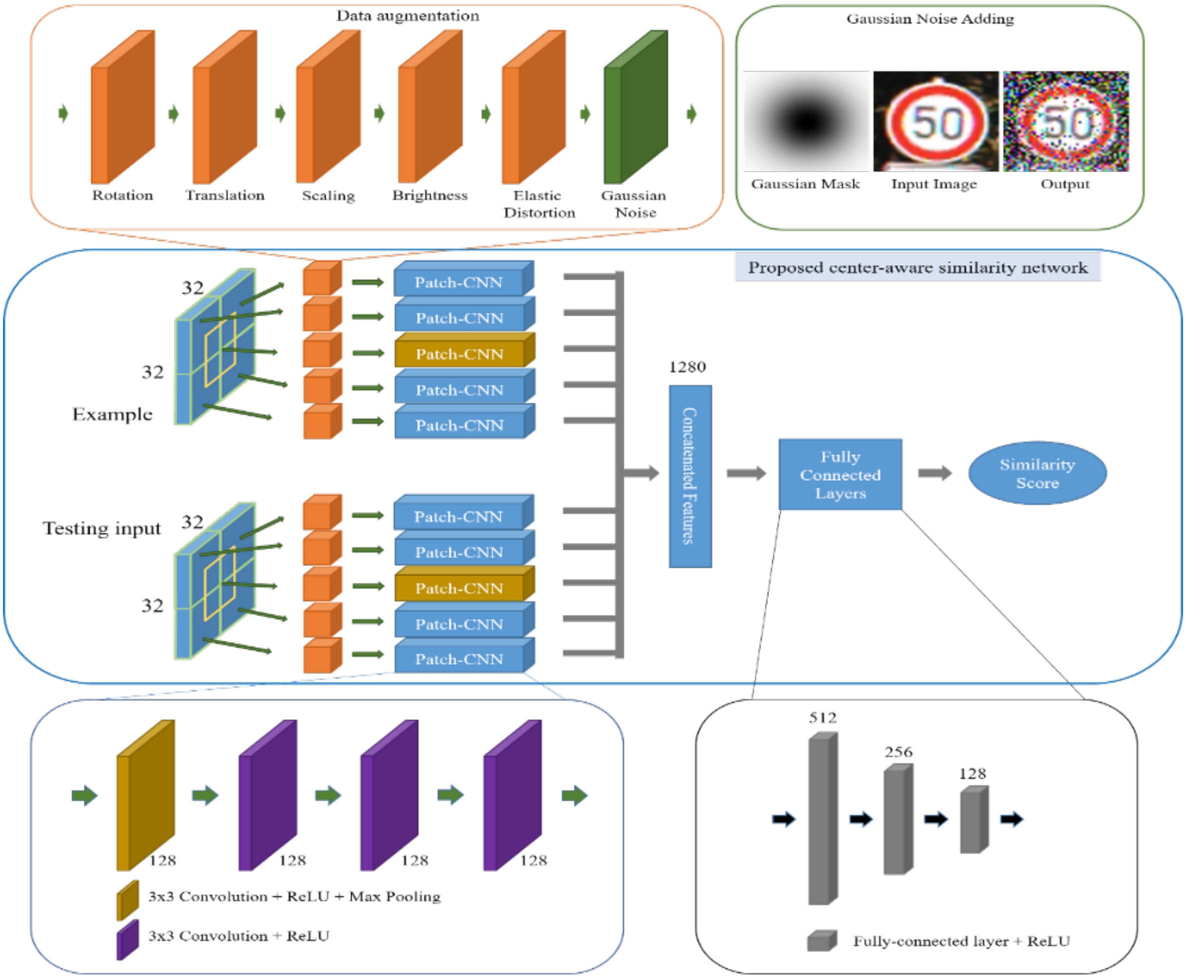

Fig. 2. Architecture of the proposed center-awareness similarity network.

The transformation pipeline for a patch to prepare a training dataset for a similarity network consists of six sub-components (Fig. 2). Gaussian noise addition in the proposed patch transformation forces the network to focus on the central part of input images. The input image is divided into five parts before feeding to the proposed network. This helps reduce computation costs while keeping performance. The central part is reserved so that the network can exploit the information of the central region of the input image effectively.

Let $\hat{y}$ denote the network output for one training example, and y the class of that training example. $y = 1$ if the example belongs to the positive class, and $y = 0$ if the example belongs to the negative class. The binary cross-entropy loss L for this example is defined as

$$L(y, \hat{y}) = ylog(\hat{y}) + (1 - y)\log{(1 - \hat{y})} \tag{4}$$

The hyperparameters of the proposed network are the number of fully connected layers, the number of units in each fully connected layer, the number of feature maps in each layer, the number of convolutional layers, the size of convolution kernels, and the size of the input patch. The details of the parameter settings for the proposed network are also included in Fig. 2.

We used 32×32 image patches as input to the network, as well as smaller patches with a resolution of 16×16 pixels. The map pooling layer and four convolutional layers comprised a 3 × 3 kernel and 128 feature maps. A 1280-length vector was formed by concatenating ten 128-length feature vectors. Subsequently, the 1280-length vector was passed through three fully connected layers with 512, 256, and 128 units each. The final fully connected layer projected the output to a single number, which is the similarity score.

3.3. KTS Dataset

Our proposed traffic sign classification method can operate without requiring a training dataset. To emphasize this ability, we prepared a Korean traffic sign classification (KTSC) dataset. We installed a camera on a car and captured videos for several hours, as shown in Fig. 3.

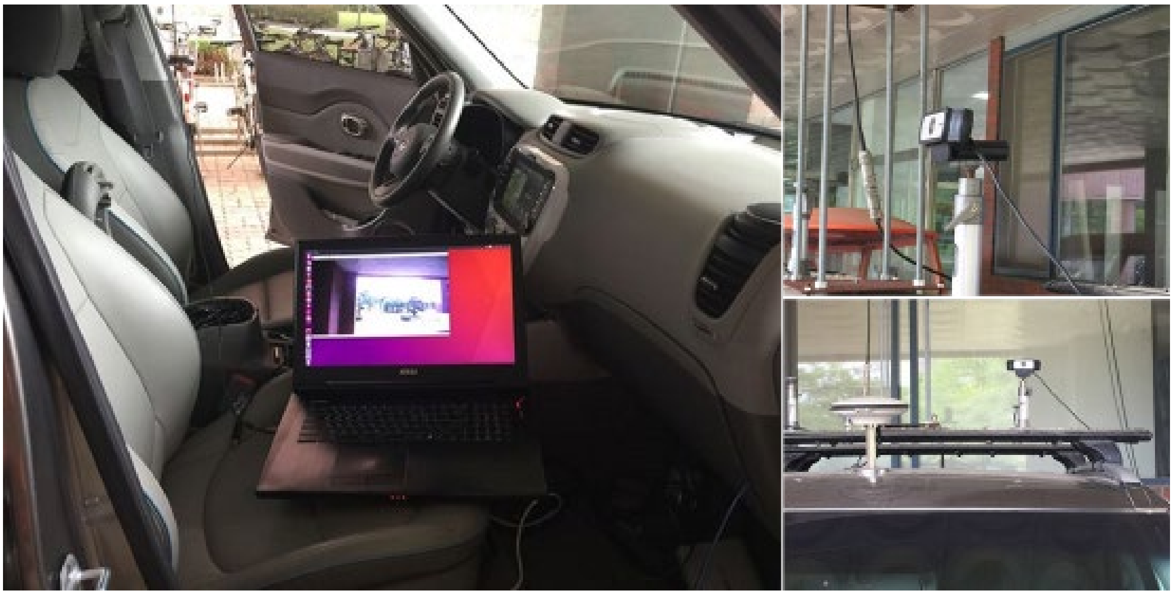

Fig. 3. Our system captures videos to prepare the Korean traffic sign classification dataset

Subsequently, we manually labeled the images extracted from the videos. The KTSC dataset included 6220 traffic sign images with 59 categories, as shown in Fig. 4 and Table 1.

KTSC does not align with the BTSC and GTSC datasets. In other words, there are categories that only appear in KTSC and vice versa. For example, GTSC has 40 classes, whereas KTSC has 59 classes. This difference clearly shows that classes (categories) of GTSC and KTSC are not the same. In addition, the traffic sign that a driver to stop his/her vehicle are visually different between the two datasets. A round shape contains the word "stop" in GTSC, while KTSC uses a around shape that contains two words "정지" and "stop" (as shown in Fig. 4).

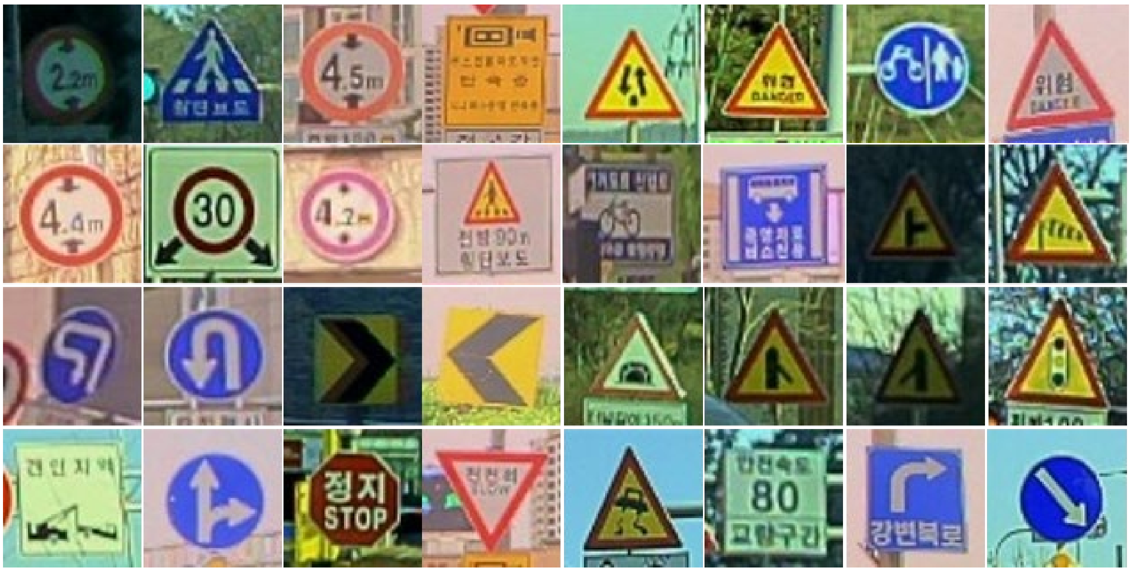

Fig. 4. Samples of Korean traffic sign classification dataset. KTSC does not align with the BTSC and GTSC datasets. There are categories that only appear in KTSC

Fig. 5. Korean traffic sign dataset with 59 categories and 6220 images. This dataset is publicly available

	Category		Category		Category
0	Bicycles and pedestrians only	20	Maximum speed limit 50	40	Pass right
1	Bicycles crossing	21	Maximum speed limit 60	41	Right curve
2	Bus only lane	22	Maximum speed limit 70	42	Right lane decrease
3	Bus sign	23	Maximum speed limit 80	43	Right turn
4	Camera sign	24	Maximum speed limit 90	44	Safe speed 80
5	Children crossing ahead	25	Minimum safe distance between vehicles	45	Slippery road
6	Crossroad	26	Minimum speed limit 50	46	Slow
7	Crosswalk	27	Motor vehicles only	47	Speed humps
8	Crosswind	28	No bicycles	48	Stop
9	End crosswalk	29	No electric car sign	49	Straight and left turn
10	End maximum speed limit	30	No entry	50	Straight and right turn
11	End of dual Carriageway	31	No left turn	51	T-shaped intersection
12	Falling rocks	32	No motorcycles	52	Tow away zone
13	Height limit	33	No overtaking	53	Traffic merge from left
14	Intersection to left	34	No right turn	54	Traffic merge from right
15	Intersection to right	35	No stopping or parking	55	Tunnel
16	Left turn	36	No trucks	56	Turn left sign
17	Maximum speed limit 100	37	No U-turn	57	Turn right sign
18	Maximum speed limit 30	38	Pass left or right	58	U-turn
19	Maximum speed limit 40	39	Yield		

Images from KTSC are captured in driving conditions. Therefore, images (in a sequence of images) are affected by outdoor factors, such as different illustrations, blurring, and difference viewpoints. Fig. 5 shows the number of samples for each class. In our work, KTSC is used as a test set. Therefore, the imbalanced distribution can be ignored in our setting.

The KTSC dataset is considered a testing dataset in our experiments, as will be presented in the next section. It is suitable for comparing our proposed method with other traffic sign classification methods because our proposed method assumes that the training dataset is not available. This is applicable because traffic signs differ based on country. Additionally, we cannot expect a training dataset to apply to all traffic signs of all countries.

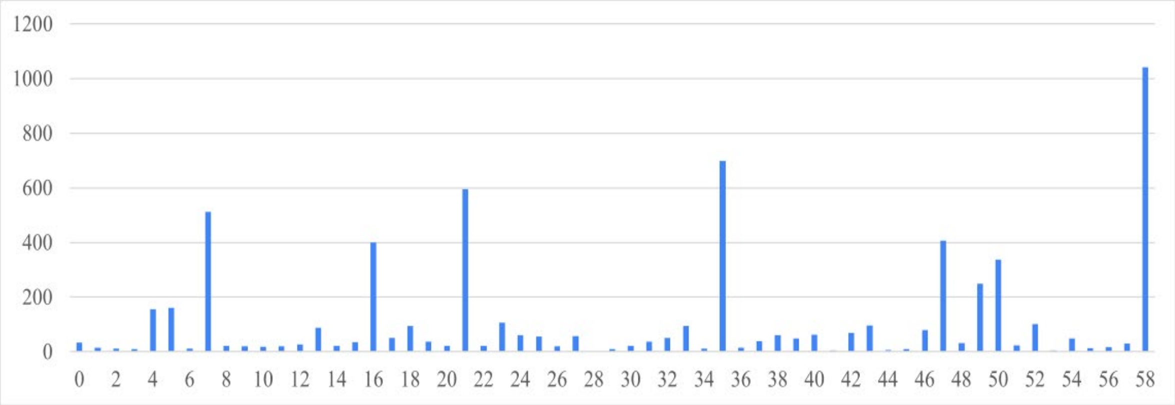

Fig. 6. The number of samples for each class in KTSC

4. Results and Conclusion

We evaluated the proposed traffic sign classification method (CASN) and compared it with traditional similarity measures such as the sum of absolute differences (SAD), normalized cross-correlation (NCC), MC-Net [4], Transformer [32], and MatchNet [33]. In addition, to demonstrate the effect of using spatial Gaussian noise, we evaluated a version of the proposed method that does not involve Gaussian noise, namely CASN(-). We used two traffic sign datasets, the GTS and BTSC datasets. We evaluated the testing method by computing top-1, -2, and -3 accuracies, where the top-3 accuracy for a method implies that the method's top three estimates contain the correct answer. Top-n accuracy indicates that the correct class is counted if it appears in top-n similarity scores. The parameter settings for the proposed method are presented in Table 2.

Table 1. Parameter setting for the image patch augmentation

Name	contrast	brightness	rotation	translation	data distance	scaling	σ	EDalpha	EDsigma
Value	[1,1.1]	[0,0.4]	[-10,10]	[-1,1]	5	[0.9,1]	[11,14]	[1,7]	[1,7]

We used the KITTI optical flow dataset [34] to construct a dataset to train our similarity network. Based on the ground truth, we extracted image patches and created approximately 34 million pairs. The network was trained using a stochastic gradient descent method to optimize the cross-entropy loss. The network was trained for 20 epochs with the learning rate initially set to 0.004 and decreased by a factor of 10 on the 15th iteration. The training examples were shuffled before learning, and the batch size was set to 128. In our following experiments (Sections 4.2 and 4.3), we intentionally added some Korean traffic signs to the BTSC and GTS testing datasets to emphasize the limitation of supervised traffic sign classification methods. The supervised classification methods failed to operate on a testing dataset that had traffic sign categories that differed from those of a training dataset.

4.1. Comparison with supervised methods

We evaluated our proposed method using SAD, NCC, MatchNet [33], MC-Net [4], and Transformer [32] traffic sign classification methods using the KTSC dataset. The MC-Net and Transformer methods are supervised approaches and were trained on the GTS dataset. Supervised methods for traffic sign classification depend on traffic sign types that are defined on a training dataset, and these supervised methods are not designed to classify traffic signs from different domains.

However, the proposed traffic sign classification method does not require a training dataset to be constructed. Therefore, it is not constrained to sign categories and applies to any type of traffic sign. The proposed method provides a few examples for each traffic sign to operate. However, it is significantly easier to obtain a few examples of each traffic sign than to acquire tens of thousands of examples for training. This is the main advantage of the proposed method compared with supervised methods.

Table 3 shows the quantitative results of the testing methods using the KTSC dataset. It is noteworthy that the KTSC dataset does not provide a training dataset and only contains a small set for evaluation. The MC-Net and Transformer trained using the BTSC dataset did not perform well on the new domain (KSTS). SAD, NCC, and the proposed method were operated with the KTSC dataset without a training dataset. MatchNet, CASN(-), and CASN performed better than MC-Net and Transformer because MC-Net and Transformer were trained in different datasets with different categories. CASN(-) can be considered an improved version of MatchNet for this specific traffic sign classification problem, as CASN(-) accounts for the center region of a patch. Consequently, CASN(-) performed better with MatchNet. Because of Gaussian noise, CASN outperformed CASN(-) in all the testing cases.

Table 2. Accuracies of the testing traffic sign classification methods using KTSC

Top-n	SAD	NCC	MC-Net	Transformer	MatchNet	CASN(-)	CASN
Top 1	13.70%	15.66%	14.78%	17.52%	42.16%	45.89%	51.01%
Top 2	17.19%	19.46%	16.92%	21.58%	48.62%	55.81%	66.39%
Top 3	21.24%	22.83%	19.64%	24.76%	51.48%	61.35%	70.02%

4.2. BTSC

The BTSC dataset was prepared for traffic sign classification purposes; it is a subset of the Belgian traffic sign dataset and includes cropped images around annotations for 62 different classes of traffic signs. The BTSC dataset comprises a training set with 4591 images and a testing set with 2534 images. Fig. 6 shows some examples of traffic signs in the BTSC dataset.

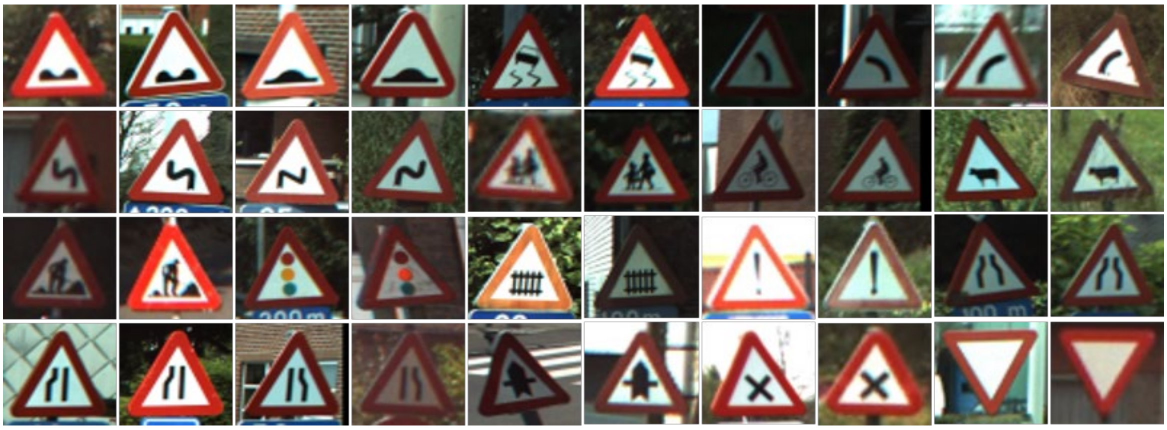

Fig. 7. Examples of traffic signs from the BTSC dataset. Important information to distinguish the traffic signs mainly located in the central regions

We evaluated the performances of SAD, NCC, MatchNet, CASN(-), and CASN using the BTSC dataset. In these experiments, MC-Net [4] and Transformer [32] were not included because we aimed to evaluate the methods above in a several-shot learning approach. For each type of traffic sign, we selected two examples from the training set. Table 4 shows the accuracies of the testing methods using the BTSC dataset for one and two examples. The simple SAD method operated poorly and yielded the worst performance. NCC was able to tolerate linear transformations between patches and performed better than the SAD. When the number of examples increased, SAD and NCC performed slightly better than those using one example. MatchNet, based on a convolutional network, performed much better than NCC.

Meanwhile, CASN(-) and CASN effectively utilized examples when the accuracies increased by more than 10% for the top-1, -2, and -3 cases. In addition, CASN indicated higher accuracies than CASN(-

) by approximately 11%, 9%, and 5% for the top-1, -2, and -3 cases, respectively. Overall, CASN performed significantly better than SAD, NCC, MatchNet, and CASN(-).

Table 3. Accuracies of the testing traffic sign classification methods using BTSC

Examples	Top-n	SAD	NCC	MatchNet	CASN(-)	CASN
1	Top 1	19.85%	39.1%	47.33%	51.18%	62.47%
1	Top 2	26.87%	48.77%	56.16%	62.62%	70.95%
1	Top 3	31.41%	54.73%	61.22%	69.06%	74.86%
2	Top 1	21.82%	43.72%	55.82%	62.50%	75.37%
2	Top 2	29.4%	52.99%	68.4%	73.48%	82.32%
2	Top 3	35.12%	59.82%	69.23%	79.16%	85.59%

4.3. GTS

In this subsection, we evaluated the performance of the testing methods using the GTS dataset. The GTS dataset comprises 39,209 color images for training and 12,630 images for testing. Each image belongs to one of the 43 classes. Fig. 7 shows some examples of traffic signs in the GTS dataset.

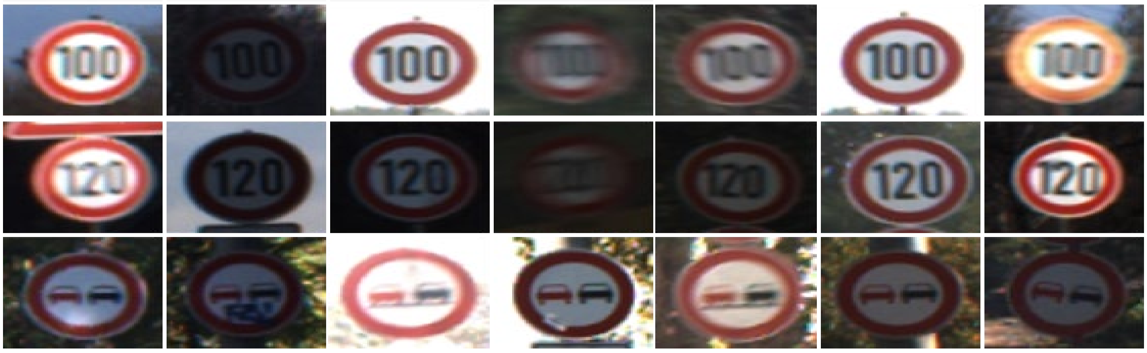

Fig. 8. Examples of traffic signs from the GTS dataset. Traffic signs are with different blurring and are captured under different conditions of illumination

For each type of traffic sign, we selected seven examples from the training set. Table 5 shows the quantitative results of the testing methods using the GTS dataset. Generally, the GTS dataset is more challenging than the BTSC dataset because the scale levels of traffic signs in the former are broader than those in the latter. Consequently, the SAD and NCC, which do not tolerate object scaling, indicated inferior performance (less than 12% for all the cases of different examples).

Table 4. Accuracies of testing traffic sign classification methods using GTS

Examples	Top-n	SAD	NCC	MatchNet	CASN(-)	CASN
1	Top 1	3.24%	4.53%	34.33%	43.56%	53.24%
1	Top 2	6.54%	7.66%	42.19%	56.45%	65.41%
1	Top 3	9.15%	10.9%	47.41%	65.03%	71.82%
2	Top 1	3.61%	3.43%	35.67%	48.10%	54.32%
2	Top 2	8.11%	7.36%	44.52%	60.38%	66.23%
2	Top 3	11.9%	10.9%	48.17%	67.66%	72.02%
3	Top 1	4.01%	2.94%	36.97%	51.05%	57.26%
3	Top 2	7.86%	6.89%	45.62%	62.89%	67.66%
3	Top 3	11.0%	10.4%	49.03%	70.11%	73.26%
4	Top 1	4.86%	4.01%	38.66%	53.18%	60.11%
4	Top 2	8.61%	7.18%	47.84%	64.84%	69.27%
4	Top 3	11.4%	9.99%	51.01%	71.25%	74.56%

Meanwhile, MatchNet, CASN(-), and CASN performed significantly better than SAD and NCC in the GTS dataset. In one example, SAD and NCC indicated average accuracies of 3.24% and 4.53%, respectively. CASN outperformed about 16 and 11 times, respectively, with an average accuracy of

53.24%. In addition, CASN using spatial Gaussian noise significantly outperformed MatchNet and CASN(-) for all cases. This proved the possible effects of using Gaussian noise in our proposed method, which was designed mainly for traffic sign classification.

4.4. Analysis of the CASN Network

In this subsection, we evaluated the effectiveness of the proposed similarity network. We compare CASN with a patch-based similarity network (PSN) trained in the same dataset and hyperparameters as CASN. The PSN is a simple version of CASN, in which the input patch is not divided into smaller patches.

Table 6 shows the accuracy of CASN and PSN using the BTSC and GTS datasets for one example. In all cases, CASN significantly outperformed PSN. This demonstrates the effectiveness of the proposed network, which divides the input patch into smaller patches to reduce the effect of image scaling and to employ better information in the center of the input patch.

Table 5. Accuracies of CASN and PSN networks with one example

	BTSC dataset		GTS dataset	
Top	PSN	CASN	PSN	CASN
Top 1	44.39	62.47	34.22	53.24
Top 2	56.15	70.95	47.81	65.41
Top 3	61.21	74.86	54.65	71.82

4.5. Computation Time

To measure the computation times of the proposed method, we used an experimental PC platform consisting of an Intel Core i77-7700 CPU 3.60 GHz × 8 and a TITAN Xp GPU card. In our proposed method, examples of traffic signs can be loaded and processed once offline to extract CNN features. In the online mode, for each input traffic sign that requires classification, the proposed method extracts the CNN feature for the input and computes the similarity with the computed CNN features of the examples.

We repeatedly measured (i.e., 100 times) the computation time required for the proposed method to classify an input. Subsequently, the computation time for input was computed by averaging 100 measurements. In our experiment, the proposed method required 0.011 milliseconds to classify a traffic sign.

5. Conclusion

We proposed a traffic sign classification method that was trained in a self-supervised manner. As the proposed method is based on a similarity network, it is not restricted by the limitations of supervised methods, where only traffic signs trained in the training dataset are applicable. Therefore, the proposed method is not limited to applications in a specific country. The experimental results indicated that the proposed method significantly outperformed popular similarity measurements and can be operated on the traffic signs of any country. The proposed traffic sign recognition method is constructed without traffic sign dataset. In other words, this method does not need domain dataset for a training step. In addition, the proposed method is not limited to prefixed classes. Therefore, this approach is especially good for applications that collecting training datasets is an obstacle. Many approaches are available for computing the similarity between patches, and similarity measures should be adapted to different applications. In the future, we will investigate similarity measures, which are based on statistics and deep networks, and evaluate them in several applications such as stereo matching, optical flow, and template matching.

Declarations

Author contribution. All authors contributed equally as the main contributors of this paper. All authors read and approved the final paper

Funding statement. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

Data and Software Availability Statements

Our source code and dataset are available at https://github.com/ComVisDinh/korean_traffic_sign.

References

[1] N. Gray *et al.* , "GLARE: A Dataset for Traffic Sign Detection in Sun Glare," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 12323–12330, Nov. 2023, doi: 10.1109/TITS.2023.3294411.

[2] J. Wang, Y. Chen, Z. Dong, and M. Gao, "Improved YOLOv5 network for real-time multi-scale traffic sign detection," *Neural Comput. Appl.*, vol. 35, no. 10, pp. 7853–7865, Apr. 2023, doi: 10.1007/s00521-022-08077-5.

[3] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, and W. Liu, "Traffic sign detection and recognition using fully convolutional network guided proposals," *Neurocomputing*, vol. 214, pp. 758–766, Nov. 2016, doi: 10.1016/j.neucom.2016.07.009.

[4] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale Convolutional Networks," in *The 2011 International Joint Conference on Neural Networks*, Jul. 2011, pp. 2809–2813, doi: 10.1109/IJCNN.2011.6033589.

[5] X. Zhang, P. Cui, R. Xu, L. Zhou, Y. He, and Z. Shen, "Deep Stable Learning for Out-Of-Distribution Generalization," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 5368–5378, 2021, doi: 10.1109/CVPR46437.2021.00533.

[6] D. Krueger *et al.* , "Out-of-Distribution Generalization via Risk Extrapolation," *Proc. Mach. Learn. Res.*, vol. 139, pp. 5815–5826, 2021, [Online]. Available at: https://arxiv.org/abs/2003.00688.

[7] Y. Shu, Z. Cao, C. Wang, J. Wang, and M. Long, "Open Domain Generalization with Domain-Augmented Meta-Learning," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 9619–9628, doi: 10.1109/CVPR46437.2021.00950.

[8] S. Rakshit, D. Tamboli, P. S. Meshram, B. Banerjee, G. Roig, and S. Chaudhuri, "Multi-source Open-Set Deep Adversarial Domain Adaptation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12371 LNCS, Springer Science and Business Media Deutschland GmbH, 2020, pp. 735–750, doi: 10.1007/978-3-030-58574-7_44.

[9] M. Mancini, Z. Akata, E. Ricci, and B. Caputo, "Towards Recognizing Unseen Categories in Unseen Domains," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12368 LNCS, Springer Science and Business Media Deutschland GmbH, 2020, pp. 466–483, doi: 10.1007/978-3-030-58592-1_28.

[10] P. Mangla, S. Chandhok, V. N. Balasubramanian, and F. Shahbaz Khan, "COCOA: Context-Conditional Adaptation for Recognizing Unseen Classes in Unseen Domains," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2022, pp. 1618–1627, doi: 10.1109/WACV51458.2022.00168.

[11] Z. Lu, Y. Yang, X. Zhu, C. Liu, Y.-Z. Song, and T. Xiang, "Stochastic Classifiers for Unsupervised Domain Adaptation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 9108–9117, doi: 10.1109/CVPR42600.2020.00913.

[12] J. Liang, D. Hu, and J. Feng, "Domain Adaptation with Auxiliary Target Domain-Oriented Classifier," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 16627–16637, doi: 10.1109/CVPR46437.2021.01636.

[13] Q. Xu, Y. Shi, X. Yuan, and X. X. Zhu, "Universal Domain Adaptation for Remote Sensing Image Scene Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–15, 2023, doi: 10.1109/TGRS.2023.3235988.

[14] Y. Zhu *et al.* , "Deep Subdomain Adaptation Network for Image Classification," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 4, pp. 1713–1722, Apr. 2021, doi: 10.1109/TNNLS.2020.2988928.

[15] C. Hu, S. Hudson, M. Ethier, M. Al-Sharman, D. Rayside, and W. Melek, "Sim-to-Real Domain Adaptation for Lane Detection and Classification in Autonomous Driving," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2022, vol. 2022-June, pp. 457–463, doi: 10.1109/IV51971.2022.9827450.

[16] L. Chen *et al.* , "Reusing the Task-specific Classifier as a Discriminator: Discriminator-free Adversarial Domain Adaptation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, vol. 2022-June, pp. 7171–7180, doi: 10.1109/CVPR52688.2022.00704.

[17] S. X. Hu, D. Li, J. Stuhmer, M. Kim, and T. M. Hospedales, "Pushing the Limits of Simple Pipelines for Few-Shot Learning: External Data and Fine-Tuning Make a Difference," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, vol. 2022-June, pp. 9058–9067, doi: 10.1109/CVPR52688.2022.00886.

[18] G. Zhang, Z. Luo, K. Cui, S. Lu, and E. P. Xing, "Meta-DETR: Image-Level Few-Shot Detection with Inter-Class Correlation Exploitation," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–12, Nov. 2022, doi: 10.1109/TPAMI.2022.3195735.

[19] Y. Lu, L. Wen, J. Liu, Y. Liu, and X. Tian, "Self-Supervision Can Be a Good Few-Shot Learner," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13679 LNCS, Springer Science and Business Media Deutschland GmbH, 2022, pp. 740–758, doi: 10.1007/978-3-031-19800-7_43.

[20] Y. Lu *et al.* , "Contour Transformer Network for One-Shot Segmentation of Anatomical Structures," *IEEE Trans. Med. Imaging*, vol. 40, no. 10, pp. 2672–2684, Oct. 2021, doi: 10.1109/TMI.2020.3043375.

[21] D. Tomar, B. Bozorgtabar, M. L. Guillaume Vray, M. Saeed Rad, and J.-P. Thiran, "Self-Supervised Generative Style Transfer for One-Shot Medical Image Segmentation," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2022, pp. 1737–1747, doi: 10.1109/WACV51458.2022.00180.

[22] H. Yang *et al.* , "Balanced and Hierarchical Relation Learning for One-shot Object Detection," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, vol. 2022-June, pp. 7581–7590, doi: 10.1109/CVPR52688.2022.00744.

[23] A. Radford *et al.* , "Learning Transferable Visual Models From Natural Language Supervision," *Proc. Mach. Learn. Res.*, vol. 139, pp. 8748–8763, 2021, [Online]. Available at: https://arxiv.org/abs/2103.00020.

[24] X. Gu, T. Y. Lin, W. Kuo, and Y. Cui, "Open-Vocabulary Object Detection Via Vision and Language Knowledge Distillation," *ICLR 2022 - 10th Int. Conf. Learn. Represent.*, pp. 1–21, 2022, [Online]. Available at: https://arxiv.org/abs/2104.13921.

[25] M. Hendriksen, M. Bleeker, S. Vakulenko, N. van Noord, E. Kuiper, and M. de Rijke, "Extending CLIP for Category-to-Image Retrieval in E-Commerce," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13185 LNCS, Springer Science and Business Media Deutschland GmbH, 2022, pp. 289–303, doi: 10.1007/978-3-030-99736-6_20.

[26] D. Jiang and M. Ye, "Cross-Modal Implicit Relation Reasoning and Aligning for Text-to-Image Person Retrieval," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, pp. 2787–2797, doi: 10.1109/CVPR52729.2023.00273.

[27] Y. Ge *et al.* , "Improving Zero-shot Generalization and Robustness of Multi-Modal Models," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, pp. 11093–11101, doi: 10.1109/CVPR52729.2023.01067.

[28] A. Sanghi *et al.* , "CLIP-Forge: Towards Zero-Shot Text-to-Shape Generation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, vol. 2022-June, pp. 18582–18592, doi: 10.1109/CVPR52688.2022.01805.

[29] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, vol. 2017-Janua, pp. 1647–1655, doi: 10.1109/CVPR.2017.179.

[30] S. Meister, "Outdoor stereo camera system for the generation of real-world benchmark data sets," *Opt. Eng.*, vol. 51, no. 2, p. 021107, Mar. 2012, doi: 10.1117/1.OE.51.2.021107.

[31] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, 2003, vol. 1, pp. 958–963, doi: 10.1109/ICDAR.2003.1227801.

[32] J. Simonsen and O. S. Jensen, "Contact quality in participation," in *Proceedings of the 14th Participatory Design Conference: Short Papers, Interactive Exhibitions, Workshops - Volume 2*, Aug. 2016, vol. 2, pp. 45–48, doi: 10.1145/2948076.2948084.

[33] Xufeng Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "MatchNet: Unifying feature and metric learning for patch-based matching," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, vol. 07-12-June, pp. 3279–3286, doi: 10.1109/CVPR.2015.7298948.

[34] M. Menze, C. Heipke, and A. Geiger, "Object Scene Flow," *ISPRS J. Photogramm. Remote Sens.*, vol. 140, pp. 60–76, Jun. 2018, doi: 10.1016/j.isprsjprs.2017.09.013.