Mathematical Biosciences and Engineering

MBE, 23 (3): 722–752.
DOI: 10.3934/mbe.2026028
Received: 12 January 2025
Revised: 25 September 2025
Accepted: 19 November 2025
Published: 06 February 2026

http://www.aimspress.com/journal/MBE

*Research article*

# TTSNet: Traffic sign recognition via a transformer by Learning Spectrogram Structural Features

Yi Deng [1,2], Ziyi Wu [1], Junhai Liu[1] and Hai Liu[3,4,]*

[1] School of Electronic and Electrical Engineering, Wuhan Textile University, Wuhan 430200, China.
[2] State Key Laboratory of New Textile Materials and Advanced Processing Technologies, Wuhan Textile University, Wuhan 430200, China.
[3] Wuhan Donghu University, No. 301, Wenhua Avenue, Jiangxia District, Wuhan, Hubei 430212, China.
[4] Central China Normal University, 152 Luoyu Road, Wuhan, Hubei 430079, China.

* Correspondence: Email: hailiu0204@gmail.com.

**Abstract:** Traffic sign recognition is crucial not only for autonomous vehicles and traffic safety research but also for multimedia processing and computer vision tasks. However, traffic sign recognition faces several challenges, such as high intraclass variability and interclass similarity in visual features and background complexity. We propose a novel invariant cue-aware feature concentration transformer (TTSNet) to effectively address these challenges. TTSNet aims to learn the invariant and core information of traffic signs. To this end, we introduce three new modules to learn the features of traffic signs: attention-based internal scale feature interaction (DLFL), cross-scale cross-space feature modulation (SSFM), and eliminating spatial and information redundancy (ESIR) modules. The DLFL module extracts invariant cues from traffic signs through feature selection based on discriminative values. The SSFM-Fusion module aggregates multi-scale information from traffic sign images by concatenating multi-layer features. The ESIR module improves feature representation by eliminating spatial and channel information redundancy. Extensive experiments showed that TTSNet achieves state-of-the-art performance on the T100K (89.1%) and CTSDB (89.97%) datasets.

**Keywords:** Feature extraction; image recognition; transformer; traffic sign recognition; computer vision

## 1. Introduction

Traffic sign recognition is a fundamental problem in the field of computer vision. Its aim is to provide highly accurate traffic sign predictions. With the increasing number of private cars on the road, traffic density has risen significantly, and traffic jams are also increasing. In China, image recognition for unmanned vehicles has received growing attention, maintaining an annual growth rate of 9.37% [1]. Figure 1 illustrates two major challenges in traffic sign recognition—occlusion and complex backgrounds—using real-world street scenes as examples. Occlusion is a common issue in urban environments, where vehicles, trees, and infrastructure elements block parts of traffic signs. In Figure 1(a), a large blue overhead sign is partially occluded by a bridge structure and a passing bus, making it harder to detect and read. Similarly, in Figure 1(b), a traffic sign is partially hidden behind tree branches, which obstructs visibility and increases the difficulty of recognition. On the other hand, complex backgrounds introduce visual noise and distractions, making it challenging for models to distinguish traffic signs from their surroundings. Figures 1(c) and 1(d) demonstrate such a challenge. In Figure 1(c), a traffic sign is positioned among dense trees and shadows, causing it to blend into the background, which makes detection more difficult. In Figure 1(d), the sign is located in a cluttered urban setting with trees, poles, and barriers, further complicating recognition.

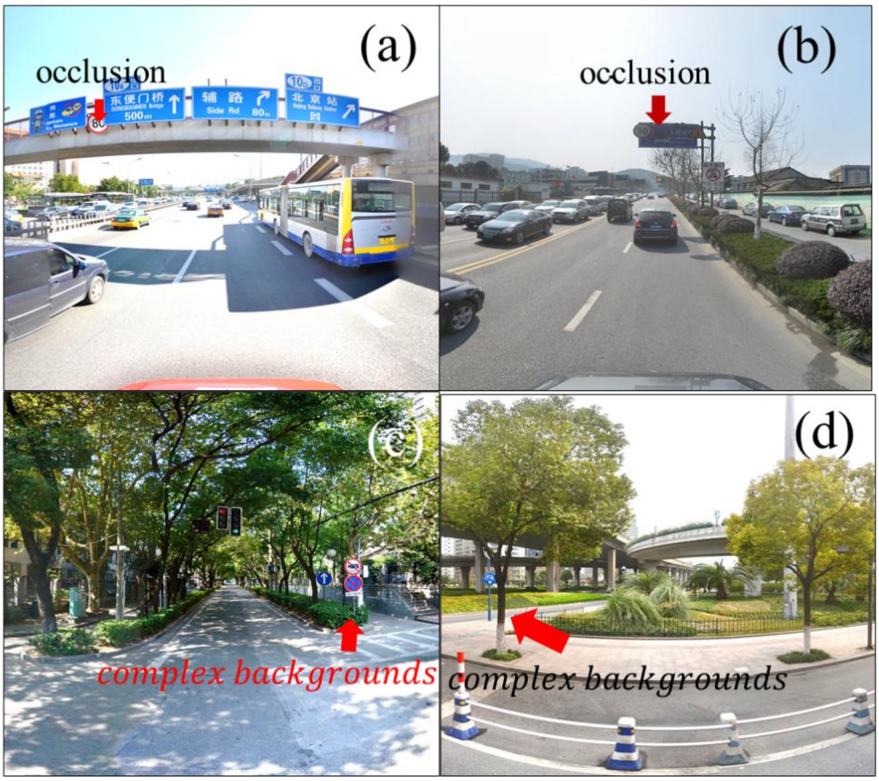

**Figure 1.** Two main challenges exist in traffic sign recognition. (a) and (b) illustrate the challenges of occlusion in traffic sign recognition. (c) and (d) demonstrate the challenges posed by complex backgrounds in traffic signs.

As autonomous vehicles are increasingly used, traffic sign recognition has become a key area of research [2,3]. This concept has attracted wide attention in the fields of multimedia processing [4,5], image classification [6,7], and image recognition [8]. However, the creation of traffic signs has long been a challenging task due to the large variation within classes and subtle variation between classes. Excellent performance on traffic signals can support downstream tasks. Automatic image recognition of traffic signs can aid a better understanding of transportation data in less developed areas. This study aims to further prevent the problem of traffic congestion and effectively maintain pedestrian safety, building on the development of recent work on traffic sign recognition.

A substantial amount of work has been carried out on traffic sign recognition. In traffic sign recognition systems, fault detection and optimization algorithms are equally important. Jawad and Abid proposed a fault detection method for HVDC systems based on the Gray Wolf Optimization algorithm and artificial neural networks, demonstrating the effectiveness and potential of optimization algorithms in complex systems  [9]. Deep learning (DL) methods have achieved better performance than traditional methods. Accordingly, we only mention DL-based image recognition methods. These can be broadly divided into two families: RoI-based methods (regions of interest) and IoU (image-only usage) methods.

For methods based on RoIs, inference relies on analyzing local regions rather than the entire image, as local areas typically provide more significant information [10,11]. These approaches commonly employ the Region Proposal Network (RPN) [12] to identify distinctive local regions. Ge et al. [10] introduced a technique where the RPN initially locates regions of interest, which are then selected, resized, and processed through a backbone network to generate valuable local features, allowing for predictions focused on these specific regions. Liu et al. [11] developed an innovative model called Filtration and Distillation Learning (FDL), which intensifies attention on discriminative areas for the task of FBIC (Fine-Grained Bi-Image Classification). FDL uniquely utilizes the alignment between proposing and predicting regions, facilitating direct optimization of the proposals. Furthermore, this approach transfers object-level knowledge to effectively enhance attention on specific regions. However, RoI-based methods may occasionally lose critical information due to the cropping of localized image sections. Additionally, the RPN backbone is limited in its ability to capture relationships among the proposed regions, leading it to often suggest larger bounding boxes that contain substantial portions of the objects rather than highlighting the most informative parts. Moreover, training models with an RPN backbone can be challenging, as optimization goals may not align. Modifying the RPN backbone also adds complexity to the overall pipeline.

IoU methods are advanced and promising because they leverage global image-level information and can be trained end-to-end without the need for additional annotations. The most widely used backbone for IoU methods is the convolutional neural network (CNN), such as VGG [13], ResNet [14], DenseNet [15], and GoogleNet [16]. Luo et al. [17] proposed an effective approach called Cross-X-Learning, which exploits the relationships between multiple images and between multiple hidden layers in the network to achieve flexible multi-scale feature learning. Cross-X offers a reasonable training time and supports continuous training with ease. It also demonstrates computational efficiency when handling large datasets. Zhuang et al. [18] introduced the Attentive Pairwise Interaction Network (API-Net), a straightforward yet effective architecture designed to recognize fine-grained distinctions by attentively identifying contrasting features between arbitrary pairs of input images. These contrasting features are obtained by computing pairwise interactions between the two images. With the addition of score ranking regularization, API-Net further generalizes its capabilities by prioritizing specific features, allowing it to be trained end-to-end as well. Du et al. [19] discovered that the key to FBIC is to encourage the network to learn at different granularities and gradually merge multi-granularity functions. Ding et al. [20] proposed an attention pyramid network for FGVC, where high-level semantic information and low-level details are exploited by building a pyramid hierarchy on a CNN. Several Transformer-based methods that can achieve state-of-the-art performance have recently been proposed due to the widespread adoption of the Transformer architecture [21]. He et al. [22] introduced the Transformer architecture for FBIC and achieved impressive performance. Their proposed model is based on vision Transformer (ViT) [23] with a novel part selection module that integrates all raw attention [24] weights of the Transformer into an attention map. Although IoU methods [25] significantly reduce the labor cost of annotating datasets and outperform previous methods, numerous challenges remain for traffic sign recognition [26].

Several challenges exist in traffic sign recognition that hinder recognition accuracy. These challenges can be summarized as follows:

1) Background complexity: Background complexity is a major problem. Traffic signs are typically located in busy urban environments where the surrounding vehicles, pedestrians, and various advertisements can interfere with recognition. This problem requires the model to not only identify the sign itself but also ignore or minimize the effects of background distractions.

2) Feature occlusion: The visual features of characters can exhibit high intraclass variation and interclass similarity. For example, the same traffic sign may appear distinct when photographed from different angles, distances, and lighting conditions, while various categories of traffic signs may overlap in shape or color. In this scenario, the model's detection ability is severely compromised, necessitating the design of network architectures that can effectively extract key features.

We have identified several important features by carefully observing the types of traffic signs and their performance in different environments. First, the shapes and colors of traffic signs vary in different traffic scenarios, but some invariant features can be used for fine-grained traffic sign classification. Second, some traffic signs may look similar but actually have varied meanings due to regional differences and different regulations. Although these characters look similar, their classification meanings are completely different.

Figure 2 shows different scenarios for traffic signs captured in various real-world environments, including urban streets and motorways. On the left-hand side of the picture are road signs in complicated conditions, such as speed limits, warning signs, and direction signs placed on overhanging structures or road signs. The right side of the picture shows how to extract and magnify the road signs to highlight their details. These enlarged images highlight specific features of the traffic signs, such as numerical speed limits, prohibition symbols, and various color-coded categories. There are therefore two lessons for the challenges set out above.

*Finding I: Invariant cues of specific traffic signs.* Traffic signs are not simply categorized by their shape or color. Certain core features, such as patterns, symbols, and text, must be differentiated from others. Analysis of images of the same sign from different angles and lighting conditions can result in misleading information that negatively affects fine-grained detection. Nonetheless, we can effectively mitigate this risk by identifying these core features and the long-term semantic relationships defined as invariant cues between graphic elements, such as the relationship between patterns and backgrounds or colors and shapes.

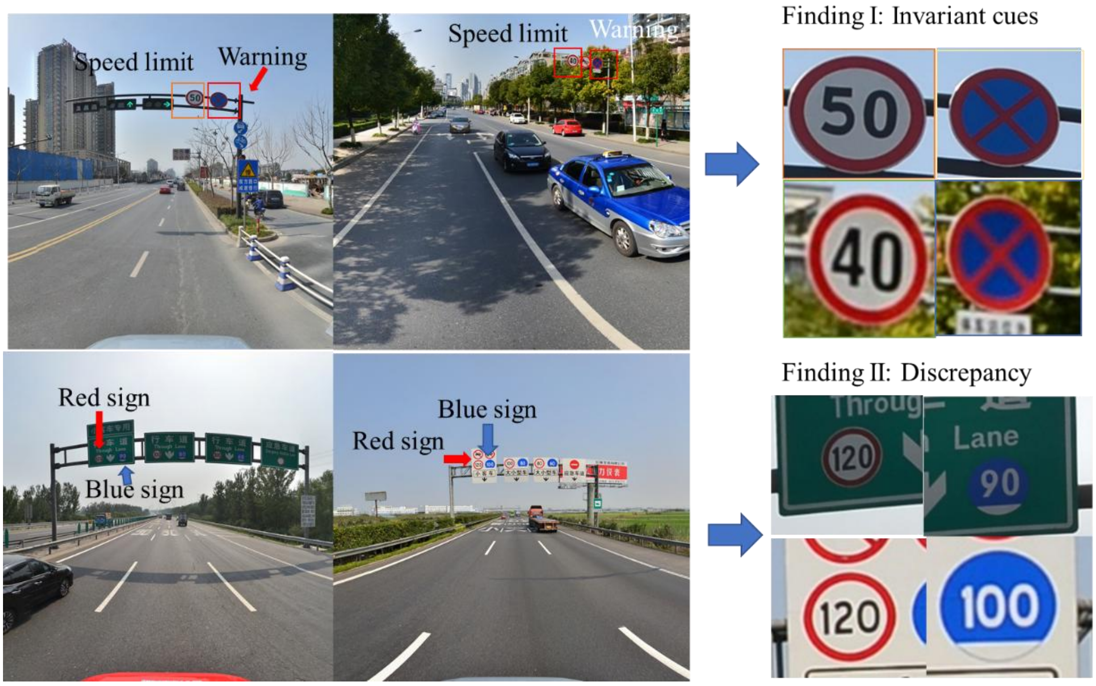

**Figure 2.** Two findings exist in traffic sign recognition.

*Finding II: Subtle discrepancies among different traffic signs.* In certain types of traffic signs, subtle differences may not be easily noticeable. However, these differences may represent completely different instructions or meanings. For example, some signs may be nearly identical in shape, but slight variations in color or border design may indicate different traffic rules. Such subtle differences are crucial for the precise recognition of traffic signs. Therefore, the recognition of crucial fine-grained features in the classification of traffic signs is of particular importance.

The above findings highlight the challenge of identifying invariant cues and subtle discrepancies in traffic sign images. Invariant cues remain consistent under varying conditions (e.g., lighting changes or occlusions), while fine-grained yet important details are often overlooked by coarse classification methods. Therefore, the effective use of these two insights plays an important role in improving the accuracy of traffic sign recognition. This work is motivated by the need to develop a method that focuses on invariant cues and enables the identification of invariant relationships between the components of traffic signs and the determination of the differences between specific traffic signs to achieve this goal. To address this, we design a multiscale feature aggregation module that integrates diverse visual information. Additionally, a feature abstraction module is introduced to extract the invariant and essential features of traffic signs. In summary, these two modules will enable our model to respond to the invariant cues in traffic sign images, thereby improving the performance of traffic sign recognition.

Unlike previous studies on traffic sign recognition that focus on exploiting common traffic sign features, our work adopts an insight perspective to uncover the invariant cues of traffic sign images. Our motivation is two-fold: On the one hand, traffic sign recognition must be able to find the invariant clues of certain traffic signs. On the other hand, distinguishing the discrepancy is crucial for recognizing similar traffic signs. We propose a novel feature transformation model capable of learning multi-scale semantic information and invariant cues in traffic signs to exploit the insights we observed. Overall, the main contributions of this work are as follows: An efficient TTSNet model was developed to exploit the results we observed in traffic sign datasets. Thereafter, a Transformer was used to study the positional relationships between traffic signs. In addition, we developed a feature extraction and fusion strategy to generate feature maps for our TTSNet model.

## 2. Related work

### *2.1. Problem formulation*

Traffic sign recognition can be briefly summarized as follows: Given a traffic sign image $x$ and its corresponding class $y$, the task is to find a mapping function $F$ to estimate $\hat{y} = F(x)$. $\hat{y}$ should fit the real traffic sign class as closely as possible. Currently, neural networks are widely utilized for the mapping function $F$. The focus of the issue lies in the network design. To minimize the loss, the network parameters $\theta$ and $F$ are updated iteratively. The optimization process typically uses gradient-based methods, such as stochastic gradient descent:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\hat{y}, \mathrm{y}), \tag{1}$$

where $\eta$ is the learning rate, and $\nabla_\theta \mathcal{L}$ is the gradient of the loss with respect to the parameters.

In recent years, numerous network architectures have emerged, such as CNNs, graph convolutional networks, and Transformers [26]. These DL-based methods establish a bridge between traffic sign images and their corresponding labels. Once the network architecture is established, the parameters in function $F$ can be obtained by minimizing the error between the predicted value $\hat{y}$ and the ground truth $y$. The distance between $y$ and $\hat{y}$ is typically measured using the mean square error (MSE), which is used to measure the difference between predicted and true values. The formula is as follows:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{2}$$

where $N$ is the total number of data points, $y_i$ is the true value (ground truth) of the i-th data point, and $\hat{y}_i$ is the predicted value for the i-th data point.

### *2.2. Transformer-based image recognition*

Transformer-based methods for computer vision can be classified into three main approaches: pure transformer architectures such as Vision Transformer (ViT), improved transformer variants such as Cross Transformer, Swine Transformer, and MSG Transformer, and hybrid models combining CNN and transformer with a focus on long-range dependency capture.

*Origin and general application*: The Transformer [27] was originally proposed for natural language processing and has since found wide application in various fields due to its exceptional ability to model long-dependent semantic relationships. Carion et al. [28] introduced this mechanism for the first time in computer vision tasks, proposing a Transformer-based object detector that achieved excellent performance. Subsequently, a novel unsupervised pretraining method [29] was presented to improve the performance of Transformer-based models.

*Transformer architectures and variants*: Dosovitskiy et al. [30] directly applied a pure Transformer architecture, showcasing its capabilities in vision tasks. Chen et al. [31] proposed CrossViT, a dual-path Transformer architecture designed to capture and integrate multiscale features for image classification. Liu et al. [32] developed a hierarchical Transformer using sliding windows, enabling self-attention computation within these windows. This approach maintains linear computational complexity and facilitates cross-window connectivity. The MSG-Transformer [33] addresses the trade-off between efficiency and flexibility by assigning messenger tokens to each region, serving as carriers for cross-regional information exchange. However, the division of images into non-overlapping segments may limit the capture of essential fine-grained details.

*Hybrid models combining Transformers with other architectures*: Liu et al. [34] combined CNN and Swin Transformer [32], where CNN was used to extract superficial features, and the Swin Transformer was employed to exploit long-dependent semantic relationships. Li et al. [35] merged CNN and Transformer architectures for human pose estimation. The integration of CNNs and Transformers addressed specific challenges in capturing fine-grained details and long-range dependencies.

### *2.3. Attention mechanism*

The attention mechanism plays a crucial role in improving the efficiency and accuracy of deep learning models. Its main function is to allow the model to focus on the most important features of the input data while ignoring irrelevant or less important information. By prioritizing essential features, the attention mechanism helps reduce noise, improve feature extraction, and improve the learning efficiency of the model. This feature is particularly useful for traffic sign recognition, where certain visual patterns, shapes, and colors are more important than others. Integrating an attention mechanism into the recognition process has been shown to significantly improve both learning efficiency and recognition accuracy [36].

In the context of traffic sign recognition, attention mechanisms have been successfully combined with traditional convolutional neural networks (CNNs) to achieve remarkable performance improvement. For example, Sun et al. [37] introduced a novel model called MobileNets CNN (MCNN), which was specifically designed for traffic sign patterns identification. This model integrates the squeeze-and-excitation (SE) module into the CNN architecture. The SE module is an advanced attention mechanism that adaptively recalibrates channel-wise feature responses, effectively improving the network's ability to focus on informative features while suppressing irrelevant ones. By integrating the SE module, MCNN improves the feature representation capability, allowing the model to better capture fine-grained details and complex patterns in traffic sign images.

Overall, the integration of attention mechanisms, exemplified by the squeeze-and-excitation module in MCNN, demonstrates the potential of combining advanced feature prioritization techniques with deep learning architectures to achieve state-of-the-art results in traffic sign recognition.

## 3. Proposed TTSNet model

### 3.1. Overview

The pipeline of the TTSNet (Traffic Sign Recognition via a Transformer by Learning Spectrogram Structural Features) model is shown in Figure 3. The proposed method includes four parts. The first part is the feature extraction backbone for extracting the multi-scale information of traffic signs. In this study, the CS module is adopted as the core of the backbone because it is excellent for eliminating spatial redundant features and information redundancy in traffic signs, such as warning signs and prohibition signs, using the shifted window scheme. The backbone has been further enhanced with the attention-based internal scale feature interaction (DLFI) for improved performance. The second part is the DLFI module, which encodes and decodes images. The third part is the SSFM fusion module (Cross-scale Cross-space Feature Modulation), which merges the multi-layer and multi-scale information of images. The fourth part is the detection head, where feature maps are finally used for the final detection.

We have prepared a glossary of abbreviations to clarify the terminology used throughout the manuscript. Table 1 lists all module names and abbreviations consistently, along with their full definitions. The glossary ensures that readers can easily understand the components of our method, avoiding confusion caused by inconsistent naming in earlier drafts.

**Table 1**. Glossary of abbreviations.

| Abbreviation | Full name |
|---|---|
| DLFI | Attention-based Internal Scale Feature Interaction |
| SSFM | Cross-scale Cross-space Feature Modulation |
| ESIR | Eliminating Spatial and Information Redundancy |
| Input-IoU | Input-guided Intersection-over-Union |
| CIoU | Complete Intersection-over-Union |
| PWC | Pointwise Convolution (1×1 Conv) |
| GWC | Groupwise Convolution |

### 3.2. ESIR module

Spatial and channel redundancy between features is exploited for TTSNet, and an efficient convolution module called ESIR is proposed to reduce redundant computing and facilitate the learning of representative features. The proposed ESIR consists of two units: spatial and channel reconstruction units. The SRU uses a separate-and-reconstruct method to reduce spatial redundancy, while the CRU uses a split-transform-and-fuse strategy to minimize channel redundancy. In addition, ESIR is a plug-and-play architectural unit that can be seamlessly integrated into various CNNs and directly replaces standard convolutional layers.

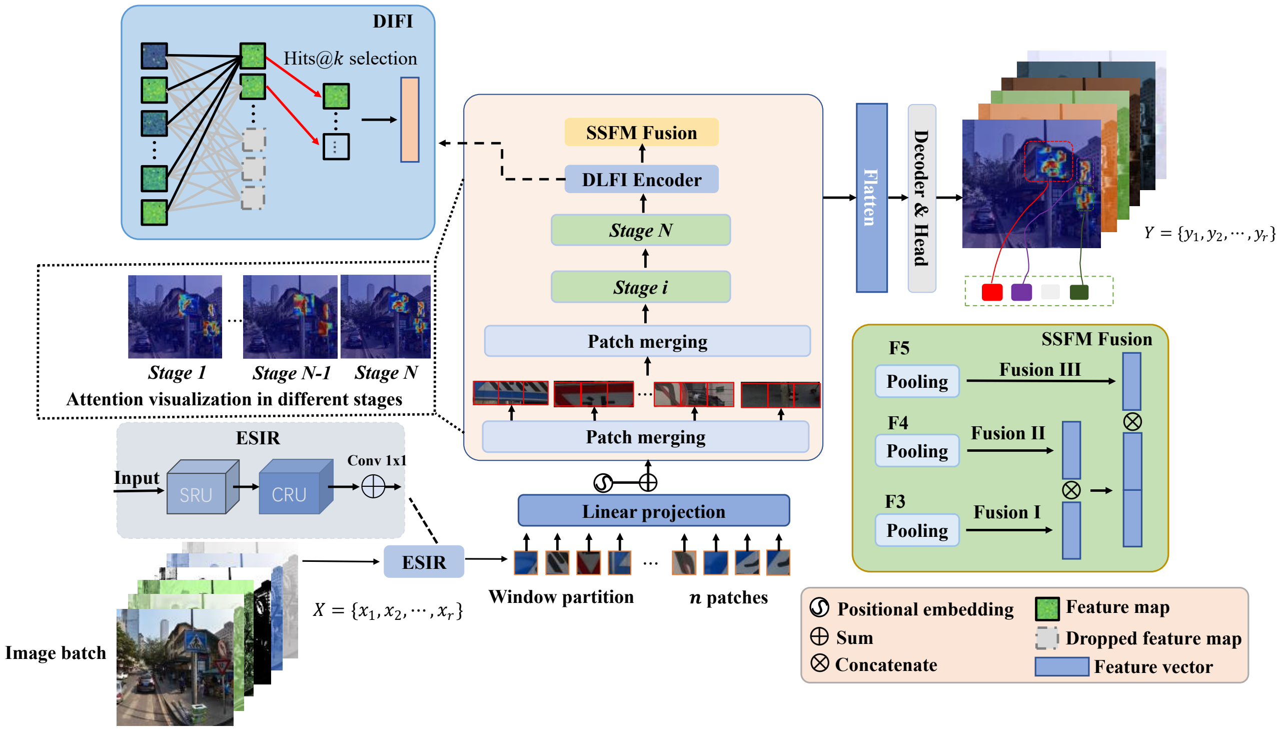

**Figure 3.** Overall structure of the TTSNet. First, the input image first passes through SRU and CRU modules to extract various features, and the extracted features are merged by 1×1 convolution. Second, the DLFI encoder further processes these features using a self-attention mechanism to capture global information and generate richer feature representations through linear projection. Third, the ESIR module processes the input image, generates multi-scale feature maps, and passes them to the subsequent multi-scale feature fusion module. Fourth, the SSFM module performs cross-scale fusion of features at different levels, and the final detection or classification result is output through the decoder.

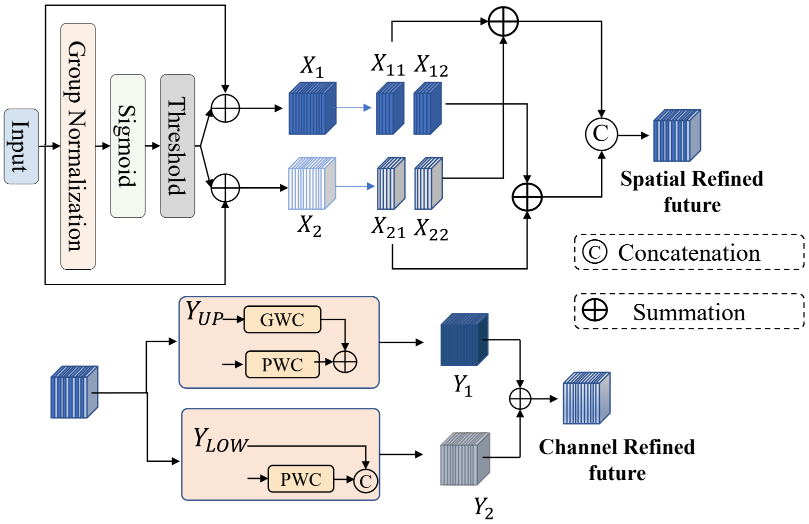

**Figure 4.** Structure diagram of the ESIR module.

The Spatial Reconstruction Department leverages a separate and reconstructed approach. The purpose of the separation operation is to distinguish feature maps with high information content from those with low information content corresponding to the spatial content. In addition, the scale factor is used in group normalization (GN) to evaluate the information content of different feature maps:

$$X_{output} = GN(X) = \gamma \frac{X-\mu}{\sqrt{\sigma^2+\varepsilon}} + \beta, \tag{3}$$

which is the transformation formula for the input data; $\mu$ and $\sigma$ represent the mean and standard deviation of $X$, respectively; and $\epsilon$ is a small positive constant added for division stability. Parameters $\gamma$ and $\beta$ are trainable affine transformations. Specifically, we exploit the trainable parameters $\gamma \in R\ C$ in the GN layers to evaluate the variance of spatial pixels for each batch and channel. The richer spatial information indicates greater variation in spatial pixels and contributes to a larger $\gamma$:

$$W = Gate(Sigmoid\left(W_\gamma\big(GN(X)\big)\right), \tag{4}$$

where the equation defines the gating mechanism used to modulate the normalized input features. The input $X$ contains informative and expressive spatial content, while $X_\omega$ captures less informative features. We first multiply the input features $X$ by $W_1$ and $W_2$, yielding two weighted features: the informative features $X_1$ and the less informative features $X_2$. This process effectively separates the input features into two distinct parts.

The Channel Reconstruction Unit (Figure 4) is introduced, utilizing a split-transform-and-fuse strategy to exploit the channel redundancy of functions. Pointwise convolution (PWC) uses 1×1 convolutions to process feature maps. PWC allows channel information to be mixed without changing the spatial dimensions. Features can be effectively combined and redundancies reduced while retaining important information by applying filters across all spatial locations for each channel. PWC is computationally efficient and helps in dimensionality reduction.

PWC, also known as 1×1 convolution, individually operates at each spatial location, shuffling information across channels without changing spatial dimensions. The formula for PWC is as follows:

$$Y_{PWC} = \sum_{l=1}^{L} W_P^{(l)} * Y^{(l-1)} + \sum_{n=1}^{N} \frac{\partial^2 Y^{(l-1)}}{\partial x_n^2}, \tag{5}$$

where $Y_{PWC}$ is the output feature map after PWC, and $W_P^{(l)}$ represents the convolution kernel for the $l$th layer, applied over the input feature map $Y^{(l-1)}$. Given that PWC is a $1 \times 1$ convolution, it changes the number of channels without modifying the spatial dimensions. $\sum_{n=1}^{N}(\partial\text{^}2\, Y\text{^}((l-1)))/(\partial x\_n\text{^}2\ )$ is a second-order derivative term that captures the curvature or higher-order changes in the feature map at the spatial position $x_n$, which increases sensitivity to nonlinear relationships. The main term $W_P^{(l)} * Y^{(l-1)}$ mixes information across all channels at each spatial location. Meanwhile, the second-order term adds fine-grained, nonlinear details to the feature extraction process.

Group-wise convolution (GWC) is a type of convolution that divides the input channels into groups and independently performs convolutions within each group. This approach reduces the number of parameters and calculations while capturing spatial features across all channels. The formula for

GWC is as follows:

$$Y_{GWC}^{(l)}(i,j) = \sum_{k \in G(i)} \left( \omega_k^{(l)} * Y^{(l-1)}(k,j) + \frac{\partial Y^{(l-1)}}{\partial k} \right) + \eta \sum_{m \in N(i)} Y^{(l-1)}(m,j), \tag{6}$$

where $Y_{GWC}^{(l)}(i,j)$ represents the output feature map at level $l$, group $i$, and spatial position $j$; $G(i)$ is the set of channels in group $i$; $\omega_k^{(l)}$ is the convolution kernel for the $k$th channel at level $l$; $Y^{(l-1)}(k,j)$ is the feature map from the previous level; the $*$ symbol represents the convolution operation; $(\partial Y^\wedge((l-1)\ ))/\partial k$ captures the gradient with respect to the channel $k$, resulting in temporal or spatial sensitivity to changes; and $\eta$ is a scaling factor that modulates the influence of the neighboring channels. The formula combines direct feature values, weighted contributions, and gradient-based refinements, which allows the model to eliminate redundant channel information while preserving key details. The gradient term $(\partial Y^\wedge((l-1)\ ))/\partial k$ ensures that regions with strong variations (such as edges or sign boundaries) receive higher attention, improving robustness in cluttered backgrounds. By incorporating a secondary neighborhood sum $\sum_{m\in=N(i)} Y^{(l-1)}(m,j)$, the model can adaptively fuse local and global information, helping to retain contextual details while reducing unnecessary channel redundancy.

GWC extends the idea of convolution by dividing the input channels into groups and performing convolutions within each group. This method reduces the number of parameters and computational effort compared with standard convolutions. GWC can capture more localized feature interactions by focusing on specific groups of channels and effectively managing redundancies while retaining critical information.

First, the separation process involves splitting the input $X$ into two different, distinct channels: the upper and lower channels. The upper channel, denoted as $X_{up}$, represents the input, which contains rich functions. This upper channel undergoes GWC and PWC processes. After these convolutions, the results are combined to output $Y_1$.

Meanwhile, the lower channel, denoted as $X_{low}$, serves as a complementary source of rich features. This lower channel is processed using PWC. Thereafter, the resulting features are merged with the original input $X$ to form $Y_2$. This step is what we refer to as the transformation process.

Finally, a merge operation is performed to merge outputs $Y_1$ and $Y_2$, resulting in the final output. This comprehensive process ensures that rich functionality and additional information are effectively combined, resulting in an improved representation of the input data.

### *3.3. DLFI module*

Figure 5 illustrates the feature maps from Transformer blocks after stage fusion, with the feature maps in the final stage arranged according to their distinction scores. As shown in Figure 5, in earlier stages, such as *Stage 1 and Stage 2*, the Hits@k features exhibit minimal similarity to one another, while features with poor scores are almost identical. In contrast, in the later stages, such as *Stage N*, the Hits@k features demonstrate greater similarity and are more highly activated, whereas the poorly scored features tend to appear noisy. Overall, features with high distinction scores across all stages prove to be more valuable than those with low scores. Building on this observation, we introduce the DLFI module to effectively utilize the information provided by these distinctive features, thereby improving the model's performance on the traffic sign recognition (TSR) task. To address the negative effects of disruptive features, the DLFI module is designed to mitigate their impact.

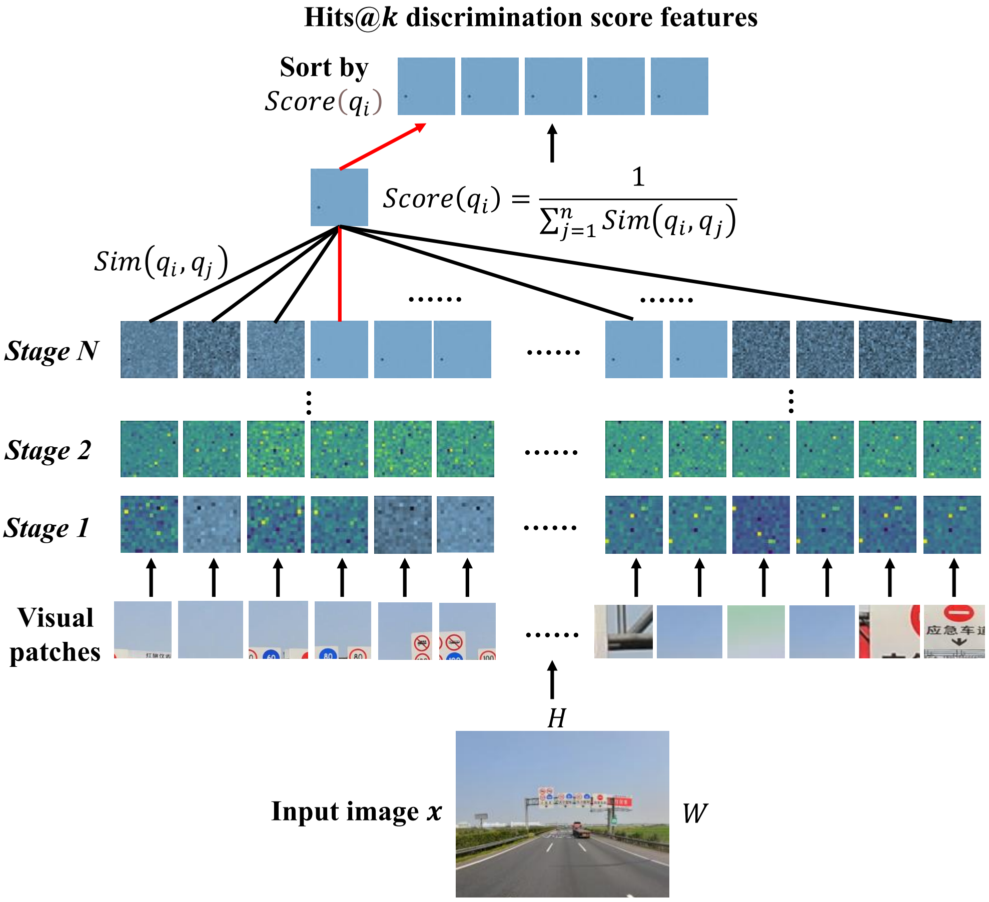
Hits@k discrimination score features
Sort by $Score(q_i)$
$Score(q_i) = \frac{1}{\sum_{j=1}^{n} Sim(q_i, q_j)}$
$Sim(q_i, q_j)$
Stage N
Stage 2
Stage 1
Visual patches
H
Input image x
W

**Figure 5.** Feature visualization in Transformer hidden layers.

First, the DLFI module calculates the similarity between $n$ vectors. This similarity can be assessed using either cosine similarity or the reciprocal of the $L_2$ distance. The cosine similarity is expressed as:

$$C_{cos}(x, y) = \frac{x \cdot y}{\| x \| \cdot \| y \|} = \frac{\sum_{i=1}^{d} x_i y_i}{\sqrt{\sum_{i=1}^{d} x_i^2} \cdot \sqrt{\sum_{i=1}^{d} y_i^2}}, \tag{7}$$

where $a \in \mathbb{R}^d$ and $b \in \mathbb{R}^d$ are two vectors, and $C_{cos}(x, y) \in [0,1]$. The value of $C_{cos}$ represents the degree of similarity between $x$ and $y$. In the DLFI module, cosine similarity is used to calculate the similarity between n vectors. Through this approach, the module can effectively capture the semantic relationships between features, thereby enhancing the model's performance in complex tasks such as occlusion handling and target recognition in cluttered backgrounds. The introduction of cosine similarity not only improves the robustness of feature representation but also provides the model with an efficient feature matching mechanism, enabling more accurate identification and differentiation of targets in high-dimensional spaces. The $L_2$distance is defined as

$$L_2(x,y) = \sqrt{\sum\nolimits_{i=1}^{d}(x_i - y_i)^2}\,, \tag{8}$$

where $a \in \mathbb{R}^d$ and $b \in \mathbb{R}^d$ represent two feature vectors. The similarity calculation is defined as

$$\begin{cases} Sim_{\cos}(v_i, qv_j) = S_{Cos}(v_i, v_j), \quad i,j \in [1,2,3,\cdots,n] \\ Sim_{L_2}(v_i, v_j) = \dfrac{1}{L_2(q_i, q_j)}, \quad i,j \in [1,2,3,\cdots,n] \end{cases}, \tag{9}$$

where $v_i$ and $v_j$ correspond to $i$-th and $j$-th patch vectors, respectively. $S_C$ represents the cosine similarity, and $L_2$ represents the $L_2$ distance.

The DLFI module uses attention mechanisms to enable effective interaction between features of different scales. This aspect is crucial for improving the model's ability to detect objects of different sizes within the same scene, which calculates interactions between features. Given a feature map $F$ with dimensions $H{\times}W{\times}C$ (height, width, and channels), the attention values can be calculated as

$$A_{i,j} = \frac{exp\ (score(F_i, F_j))}{\sum_{k=1}^{N} exp\ (score(F_i, F_k))}, \tag{10}$$

where the DLFI module combines feature maps $F_1, F_2, \ldots F_n$ from different scales to generate a unified feature representation. Let the dimensions of each feature map be $H^i \times W^i$. The merger can be expressed as

$$F_{fused} = \sum_{i=1}^{n} \alpha_i \times F_i, \tag{11}$$

where $\alpha_i$ denotes the fusion weights, which are typically learned during training, reflecting the importance of the feature map of each scale.

### *3.4. SSFM fusion module*

This network structure is designed to process input features through two parallel paths, allowing the local and global features to be captured. The first path applies a simple 1×1 convolution operation to the input, mainly focusing on simple feature transformation or dimensionality reduction (Figure 3). However, the second method is more complex as it first applies a 1×1 convolution and then passes the output through a RepBlock (re-parameterizable block) module. This module is likely to consist of a number of layers or operations that enable deeper feature extraction. Once both paths have processed their respective inputs, the outputs are combined through an element-wise summation operation (⊕). This fusion step integrates the information from both paths and allows the network to leverage simple and deeper feature representations. Such a design improves the network's ability to extract robust and comprehensive features, which can improve the overall performance of the model, especially in tasks that require multi-level feature analysis.

The SSFM fusion module in the provided diagram is a critical aspect of the entire network. In particular, this module is responsible for combining features from multiple scales or layers. The fusion process is applied in several steps. Each feature map from the different layers ($F_3$, $F_4$, and $F_5$) is processed and merged by sequential "fusion" blocks. These blocks include various operations, such as element-wise addition, concatenation, or other forms of mixing, to summarize the spatial and contextual information from each scale. This module merges three feature maps, effectively combining different types of information. Although this module can preserve various details, it also allows for an extensive, comprehensive understanding of contextual information.

$$F_5 = Reshape(DLFI(Q, K, V)), \tag{12}$$

where reshape represents restoring the shape of the flattened feature to the same shape as S5. In the overall fusion process, yellow represents a 1 × 1 convolution kernel, while blue denotes a 3×3 convolution kernel, which can be regarded as a downsampling operation, thereby reducing the size of the feature map by half. Accordingly, F5 first undergoes a 1×1 convolution and is then combined with $F_4$. The result of this combination is passed through another 1×1 convolution and combined with $F_3$. This step completes a top-down process, in which the first output result is obtained after two mergers:

$$Output = SSFM(F_3, F_4, F_5). \tag{13}$$

### *3.5. IoU loss function*

In traffic sign recognition, certain environments can be particularly complex, resulting in various traffic signs being obscured and difficult to identify, which poses challenges to the entire recognition process. A novel loss function called Input-IoU is considered to improve the detection accuracy and speed up the overall prediction regression (Figure 6).

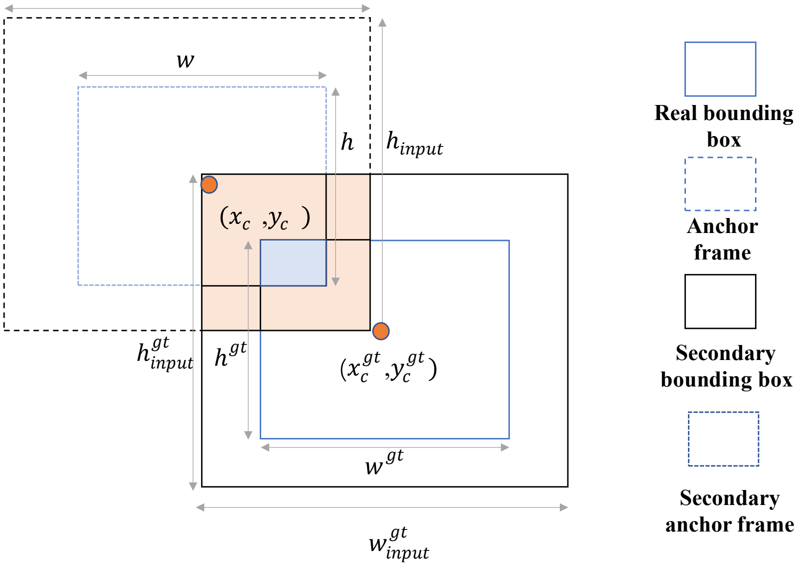

**Figure 6.** Working principle of IoU loss function.

The objective is to address the problems of slow convergence and weak generalization of the CIoU loss function in traffic sign recognition tasks. Accordingly, this work proposes the use of additional bounding boxes to calculate the loss and speed up the bounding box regression. This method controls the aspect ratio of the auxiliary bounding boxes by adding a scaling factor using different scales of the auxiliary bounding boxes for the dataset and the detector, thereby overcoming the weak generalization limitations of the current method. The ground truth bounding box and the anchor box are denoted as $b^{gt}$ and $b$, respectively. The center point of the ground truth bounding box is represented as $(x_c^{gt}, y_c^{gt})$, while the center point of the anchor box is denoted as $(x_{c,}y_c)$. The width and height of the ground truth bounding box are represented as $w^{gt}$ and $h^{gt}$, respectively. Meanwhile, the width and height of the anchor box are denoted by www and $h$, respectively. The range of the scaling factor is between 0.5 and 1.5, with the following calculation formula:

$$IoU^{input} = \frac{input}{union}, \tag{14}$$

where $IoU^{input}$ represents the intersection over union (IoU) of the input IoU, “input” refers to the intersection area between the auxiliary anchor box and the auxiliary bounding box, and “union” denotes the union area of the auxiliary anchor box and the auxiliary bounding box. Equation (15) defines the input IoU as the ratio of the intersection area to the union area between the auxiliary anchor box and the auxiliary bounding box. Unlike traditional $L_1$ or $L_2$ losses, which minimize coordinate differences, optimizing IoU directly improves the overlap between predicted and ground truth boxes. This approach enhances bounding box prediction by being scale-invariant, providing more meaningful optimization directions, and ensuring stable gradients even when boxes are highly overlapping. As a result, IoU-based optimization leads to more accurate and robust object detection. The left and right boundaries of the auxiliary bounding box are expressed as

$$b_l^{gt} = x_c^{gt} - \frac{w^{gt} * r}{2}, b_r^{gt} = x_c^{gt} + \frac{w^{gt} * r}{2}, \tag{15}$$

where $b_l^{gt}$ represents the x coordinate of the left boundary of the auxiliary bounding box, $b_r^{gt}$ represents the x coordinate of the right boundary of the auxiliary bounding box, and $r$ is the scaling factor that controls the size of the auxiliary bounding box. The top and bottom boundaries of the auxiliary bounding box are expressed as

$$b_t^{gt} = y_c^{gt} - \frac{h^{gt} * r}{2}, b_b^{gt} = y_c^{gt} + \frac{h^{gt} * r}{2}, \tag{16}$$

where $b_t^{gt}$ represents the y coordinate of the lower boundary of the auxiliary bounding box, while $b_b^{gt}$ represents the y coordinate of the upper boundary of the auxiliary bounding box. The left and right boundaries of the auxiliary anchor box are expressed as

$$b_l = x_c^{gt} - \frac{w * r}{2}, b_r = x_c + \frac{w * r}{2}, \tag{17}$$

where $b_l$ represents the x coordinate of the left boundary of the auxiliary anchor box, and $b_r$ denotes the x coordinate of the right boundary of the auxiliary anchor box. The top and bottom boundaries of the auxiliary anchor box are expressed as

$$b_t = y_c - \frac{h * r}{2}, b_b = y_c + \frac{h * r}{2}, \tag{18}$$

where $b_t$ represents the y coordinate of the lower limit of the auxiliary anchor box, and $b_b$ represents the y coordinate of the top boundary of the upper limit auxiliary anchor box. The final loss function calculation formula is as follows:

$$\begin{cases} L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{d^2} + \alpha v \\ v = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2 \\ \alpha = \frac{v}{(1 - IoU) + v} \end{cases}, \tag{19}$$

where $L_{CIoU}$ represents the CIoU loss function, $\rho^2(b, b^{gt})$ denotes the Euclidean distance between the two, $d$ is the diagonal of the smallest bounding box, $v$ is the weighting parameter, and $\alpha$ is the aspect ratio consistency parameter.

To account for the diversity of input feature distributions, we introduce auxiliary boxes to guide the learning process:

$$L_{Input-IoU} = \sum_{k=1}^{N} \omega_k \cdot L_{CIoU}(B_{P,}B_{aux}^k), \tag{20}$$

where $B_{aux}^k$ is the $k$-th auxiliary box generated from the input features. The weighting coefficient for the $k$-th auxiliary box is constrained such that $\sum_{k=1}^{N} \omega_k = 1$.

## 4. Experimental results and analysis

### *4.1. Experimental setup*

Two datasets, namely, T100K and CTSDB, are introduced for training and evaluating our TTSNet model. The T100K dataset is a large traffic sign dataset with over 100,000 annotated images, mainly used for training and evaluating object detection and recognition models. CTSDB is a dataset specifically designed for traffic sign detection and recognition in China. This dataset contains thousands of annotated images of Chinese traffic signs taken under real driving conditions.

In this paper, some representative methods used for comparison in the experiments are introduced to evaluate the performance of our TTSNet from multiple perspectives. Detection Transformer (DETR) [38] is an object detection model that uses a Transformer to directly predict bounding boxes and class labels. This model simplifies the detection process and efficiently performs across various tasks. D-DETR [39] is an improved version of DETR that uses deformable attention to focus on key areas in an image, improving efficiency and accuracy, especially for small objects, while accelerating training. Conditional (C)-DETR [40] enhances object detection by incorporating conditional reasoning, allowing the model to consider the context of objects within scenes for improved accuracy. Detector

(SSD) [41] is a fast object detection model that detects objects in a single pass using multiple feature maps for different sizes, enabling real-time performance. Salience (S)-DETR [42] improves accuracy and efficiency by integrating attention mechanisms that focus on visually important areas, particularly when detecting small or obscured objects. Faster R-CNN [43] is a deep super-resolution network (DSR-NET) that focuses on improving the image resolution. DSRNet [44] enhances scene understanding by leveraging dynamic spatial reasoning to capture complex object interactions in visual tasks. RepLKNet [45] enhances feature extraction, particularly for fine-grained details. It combines CNN and Transformer strengths, achieving superior performance in tasks like image classification and object detection. DINO [46] uses knowledge distillation between two networks (a student and a teacher) to optimize feature learning and achieves state-of-the-art results on various vision tasks, especially in unsupervised pretraining. RepViT [47] divides an image into fixed-size patches, treats each patch as a sequence, and converts them into vectors through embedding. In TTSNet, based on the empirical observation of traffic recognition, the invariant cues-aware model has two novel modules: The DLFI module, which aggregates the multiscale information of the traffic recognition, and the TTS module, which extracts the core features of traffic recognition. In TTSNet+, based on TTSNet, the max pooling operation is altered by the DLFI module to further extract core representations in multiple stages. In this way, the dimension reduction of input vectors and the concentration of important features are accomplished at the same time.

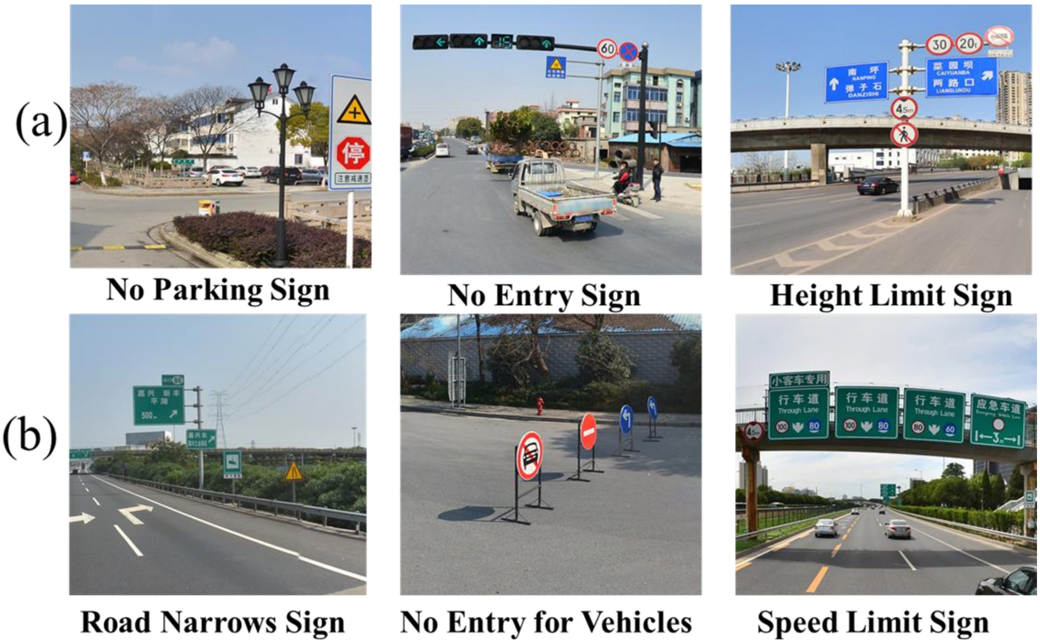

**Figure 7.** Partial displays of the T100K and CTSDB datasets. (a) T100K dataset. (b) CTSDB dataset.

The mAP refers to the mean average precision, which measures the detection accuracy of an object detection model across multiple categories. @0.5 indicates that the IoU threshold is 0.5. If the IoU between the predicted and the ground truth boxes is greater than or equal to 0.5, then the detection is considered correct. The formula for IoU is defined as follows:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}. \tag{23}$$

The *mAP*@0.5 calculates the average precision for all categories when IoU ≥ 0.5. *mAP* is calculated as the average of the average precision (AP) across all categories:

$$\mathrm{mAP@0.5} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{AP_i}. \tag{24}$$

The formula for *mAP*@0.5:0.95 is expressed as follows:

$$mAP@0.5{:}\,0.95 = \frac{1}{10}\sum_{t=0.5}^{0.95}\frac{1}{N}\sum_{i=1}^{N} AP_i(t). \tag{25}$$

where $t$ is the IoU threshold, and $APi(t)$ is the AP for the $i$th category at a given IoU threshold $i$. *mAP*@0.5:0.95 is more demanding in terms of model accuracy because it evaluates the model over a wider range of IoU thresholds.

### *4.2. Experimental result analysis*

Our method was compared with several state-of-the-art methods, and the performance of different methods was analyzed. The best value is highlighted in bold.

*1)* Results on the T100K dataset: We conducted comparative experiments on the T100K dataset, evaluating several state-of-the-art object detection models, as shown in Table 2. The quantitative results in Table 1 highlight the superior performance of our Transformer-based model, TTSNet. This advantage is largely attributed to its ability to capture long-range semantic dependencies, refine discriminative features, and enhance spatial-semantic fusion. Several benchmark models exhibit strong capabilities in specific areas: Deformable DETR efficiently handles complex scenes with varying object sizes using its deformable attention mechanism, improving speed and accuracy. DINO leverages an advanced self-supervised learning framework to achieve high recognition performance, particularly excelling with limited labeled data. SSD prioritizes real-time performance by detecting objects at multiple scales using feature pyramids, while Salience DETR integrates saliency maps with Transformers to focus on critical regions in images. Faster R-CNN balances speed and accuracy with its region proposal network, whereas DSR-Net and RepViT further optimize efficiency by refining network architectures and training strategies for real-time applications. TTSNet outperforms all baseline models, achieving an mAP@0.5 of 89.10%, which represents a 2.69% improvement over the best competing method. This superior performance is attributed to three key innovations in TTSNet; unlike DETR and DINO, which rely on standard Transformer self-attention, DLFI explicitly integrates long-range contextual relationships while enhancing discriminative features. This enables better object distinction in dense and cluttered scenes. Compared to RepLKNet and Deformable DETR, which primarily refine local receptive fields, ESIR enhances semantic feature interactions, ensuring that key object attributes are preserved while filtering out irrelevant background noise. This improves robustness in complex environments. While SSD and Faster R-CNN rely on traditional multi-scale feature fusion, SSFM dynamically adapts feature modulation, ensuring that spatial and semantic representations are optimally aligned. This leads to more precise localization and better scale adaptability.

**Table 2.** Performance comparison between our TTSnet and the state-of-the-art models on the T100k.

| Methods | Year | Foundational models | Backbone | *mAP@*.5 (%) | *mAP@*.5:95 (%) |
|---|---|---|---|---|---|
| DETR [38] | 2020 | Transfomer | ResNet-50 | 57.30 | 40.3 |
| D-DETR [39] | 2021 | Transfomer | ResNet-50 | 61.90 | 41.20 |
| C-DETR [40] | 2022 | Transfomer | ResNet-50 | 72.3 | 64.8 |
| SSD [41] | 2016 | CNN | MobileNet | 74.24 | 58.37 |
| S-DETR [42] | 2022 | Transfomer | ResNet-50 | 70.30 | 62.5 |
| Faster R-CNN [43] | 2016 | CNN | ResNet-101 | 73.36 | 66.52 |
| DSR-Net [44] | 2022 | Transfomer | ResNet-50 | 81.28 | 62.97 |
| RepLKNet [45] | 2022 | CNN | ResNet-50 | 75.1 | 67.2 |
| DINO [46] | 2023 | Transfomer | Swin | 70.70 | 51.78 |
| RepViT [47] | 2024 | Transfomer | Transformer<br>ViT | 87.79 | 66.98 |
| YOLOv5 | 2020 | Transfomer | ResNet-50 | 87.95 | 67.83 |
| YOLOv8 | 2024 | Transfomer | ResNet-50 | 88.5 | 70.2 |
| TTSNet | 2025 | Transfomer | ResNet-50 | 89.00 | 73.10 |
| TTSNet+ | 2025 | Transfomer | ResNet-50 | **89.10** | **73.20** |

*2)* Results on the CTSDB dataset: The proposed method was also evaluated on the CTSDB dataset. The corresponding results are shown in Table 3. The CTSDB still achieves an impressive *mAP@*.5 (%) of 89.97%. Additionally, RepViT consistently demonstrates exceptional performance, highlighting the importance of the end-to-end discovery approach to CTSDB tasks. Our approach, which goes one step further by leveraging multiscale information in CTSDB, is proven to increase accuracy. TTSNet shows a remarkable improvement of 1.06% over the dataset, proving the validity of our observation regarding the key differentiators.

*3)* In the comparison experiments of different loss functions on the T100K dataset, we evaluated the performance of three loss functions—CIoU, EIoU, and Inner-CIoU—across three different input resolutions: 224 × 224, 384 × 384, and 448 × 448. The experimental results are presented for each resolution, focusing on two key performance metrics: *mAP@*.5 and *mAP@*.5:95. At the 448 × 448 resolution, we compared the performance of the three loss functions from both metrics. Inner-CIoU outperformed the other loss functions significantly, achieving *mAP@*.5 and *mAP@*.5:95 scores of 89.10% and 73.20%, respectively. This indicates that Inner-CIoU is particularly effective in improving object detection accuracy. In contrast, EIoU and CIoU showed slightly lower performance, with *mAP@*.5:95 scores of 72.20% and 72.10%, respectively. Despite this, both loss functions still achieved relatively high detection accuracy. Overall, Inner-CIoU demonstrated the best generalization ability and robustness on the T100K dataset, making it the most effective choice for enhancing object detection performance among the three loss functions tested.

**Table 3.** Performance comparison between our TTSnet and the state-of-the-art models on the CTSDB.

| Methods | Year | Foundational models | Backbone | *mAP@*.5 (%) | *mAP@*.5:95 (%) |
|---|---|---|---|---|---|
| DETR [38] | 2020 | Transformer | ResNet-50 | 59.1 | 41.8 |
| D-DETR [39] | 2021 | Transformer | ResNet-50 | 63.70 | 42.20 |
| C-DETR [40] | 2022 | Transformer | ResNet-50 | 74.5 | 66.1 |
| SSD [41] | 2016 | CNN | MobileNet | 75.21 | 58.70 |
| S-DETR [42] | 2022 | Transformer | ResNet-50 | 70.70 | 63.4 |
| Faster R-CNN [43] | 2016 | CNN | ResNet-101 | 74.24 | 69.59 |
| DSR-Net [44] | 2022 | Transformer | ResNet-50 | 85.36 | 67.90 |
| RepLKNet [45] | 2022 | CNN | ResNet-50 | 76.1 | 69.2 |
| DINO [46] | 2023 | Transformer | Swin | 73.69 | 52.88 |
| RepViT [47] | 2024 | Transformer | Transformer ViT | 88.91 | 67.98 |
| YOLOv5 | 2020 | Transformer | ResNet-50 | 88.98 | 69.1 |
| YOLOv8 | 2024 | Transformer | ResNet-50 | 89.2 | 71.1 |
| TTSNet | 2025 | Transformer | ResNet-50 | 89.92 | 74.00 |
| TTSNet+ | 2025 | Transformer | ResNet-50 | **89.97** | **74.30** |

**Table 4.** Comparison experiments of different loss functions on the T100k dataset.

| Resolution | Loss function | *mAP@*.5 (%) | *mAP@*.5:95 (%) | Recall |
|---|---|---|---|---|
| 224 × 224 | CIoU | 81.1 | 63.1 | 60.5 |
| | EIoU | 81.9 | 64.3 | 62.1 |
| | Input-CIoU | **82.10** | **65.1** | **62.8** |
| 384 × 384 | CIoU | 85.4 | 67.2 | 67.5 |
| | EIoU | 85.7 | 67.5 | 67.9 |
| | Input-CIoU | **86.3** | **68.3** | **68.0** |
| 448 × 448 | CIoU | 88.2 | 72.20 | 69.2 |
| | EIoU | 88.7 | 72.10 | 70.5 |
| | Input-CIoU | **89.10** | **73.20** | **70.9** |

### *4.3. Parameter sensitivity analysis*

Various factors can influence the effectiveness of pattern recognition during model construction and training. Among these factors, the number of feature decompositions has a direct influence. We conducted comparative experiments with different numbers of feature decompositions (Figure 8) and used *mAP@*0.5, *mAP*@0.95, and recall as evaluation metrics. Therefore, the results show that the accuracy peaks when decomposing three features.

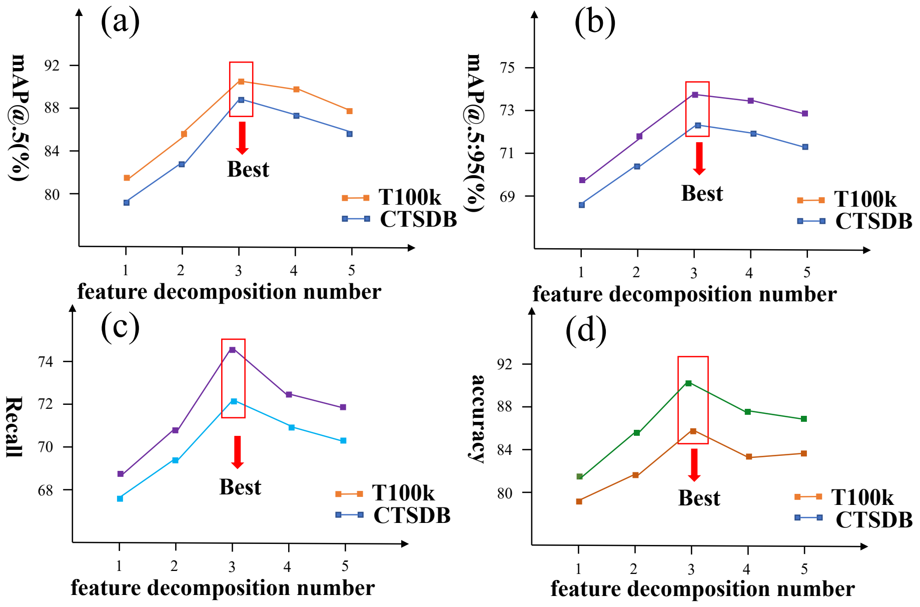

**Figure 8.** Feature decomposition number on TTSNet.

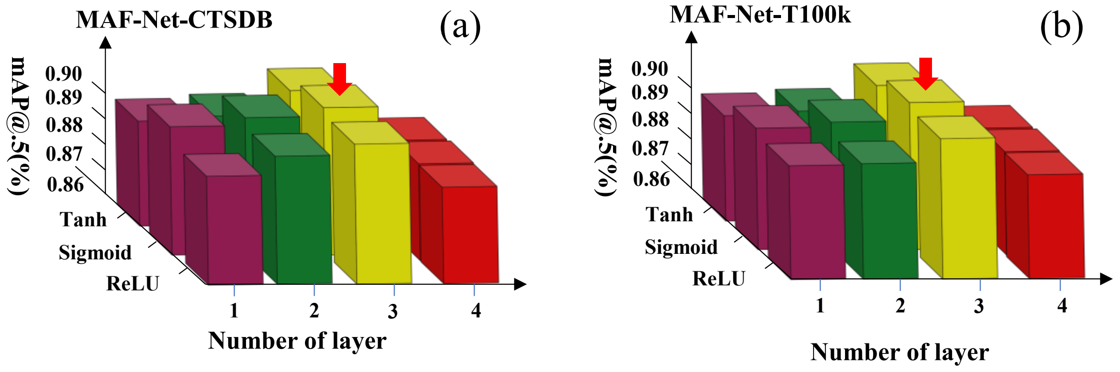

**Figure 9.** Performance of the different network depths and activation functions in the TTSNet model on the T100K and CTSDB datasets.

**Table 5.** Comparative experimental graph of the different activation functions.

| Datasets | Layer | Tanh | Sigmoid | ReLU | *mAP*@.5 (%) |
|---|---|---|---|---|---|
| T100K | 3 | ✓ | × | × | 88.30 |
| | 3 | × | ✓ | × | 89.10 |
| | 3 | × | × | ✓ | **88.70** |
| CTSDB | 3 | ✓ | × | × | 88.95 |
| | 3 | × | ✓ | × | 89.97 |
| | 3 | × | × | ✓ | **89.20** |

The results of the TTSNet model running on the T100K dataset are discussed for the activation functions (Tanh, Sigmoid, and ReLU) (Figure 9 and Table 5). The different number of network layers that we can see in the red markings shows that the best results are obtained for three network layers, as well as for the activation function of the sigmoid. Overall, the results also confirm our previous experimental results; the network with three layers and using the sigmoid activation function performed best on the CTSDB dataset.

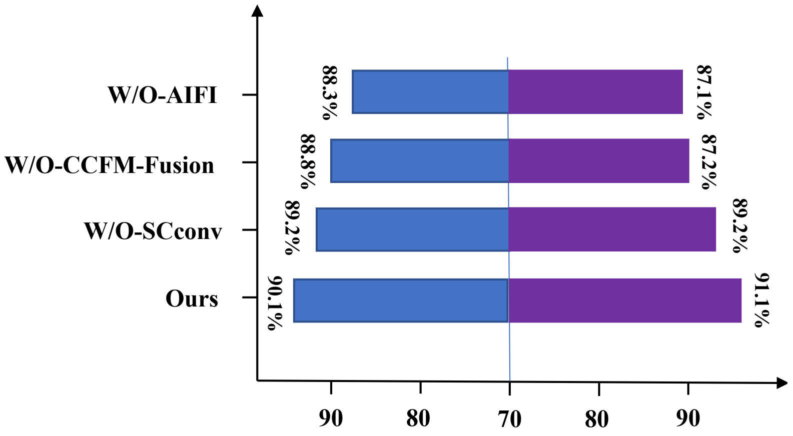

**Figure 10.** Results of the ablation experiments.

**Table 6.** Ablation experiments on the T100K and CTSDB datasets.

| Dataset | DLFI | CCFM | ESIR | Precision | *mAP*@.5 (%) | *mAP*@.5:95 (%) |
|---|---|---|---|---|---|---|
| T100K | × | ✓ | ✓ | 88.3 | 73.5 | 61.7 |
| | ✓ | × | ✓ | 88.8 | 80.1 | 65.2 |
| | ✓ | ✓ | × | 89.2 | 85.2 | 68.1 |
| | ✓ | ✓ | ✓ | **90.1** | **89.91** | **73.20** |
| CTSDB | × | ✓ | ✓ | 87.1 | 74.1 | 63.0 |
| | ✓ | × | ✓ | 87.2 | 81.1 | 66.1 |
| | ✓ | ✓ | × | 89.2 | 86.3 | 69.9 |
| | ✓ | ✓ | ✓ | **91.1** | **89.97** | **74.30** |

### *4.4. Comparison of the ablation experiments*

The effects of the DLFI, SSFM fusion, and ESIR modules were analyzed through ablation studies on the T100K and CTSDB datasets, as shown in Table 6 and Figure 10. The results indicate that removing any of these modules leads to a performance drop, demonstrating their critical contributions. DLFI enhances object distinction by capturing long-range dependencies, SSFM ensures adaptive feature fusion for better localization and scale awareness, and ESIR improves feature interactions by preserving key semantic details while reducing noise. The performance gains of 0.9% on T100K and

1.9% on CTSDB confirm that TTSNet benefits from the synergy of these three modules, leading to more robust and accurate object detection.

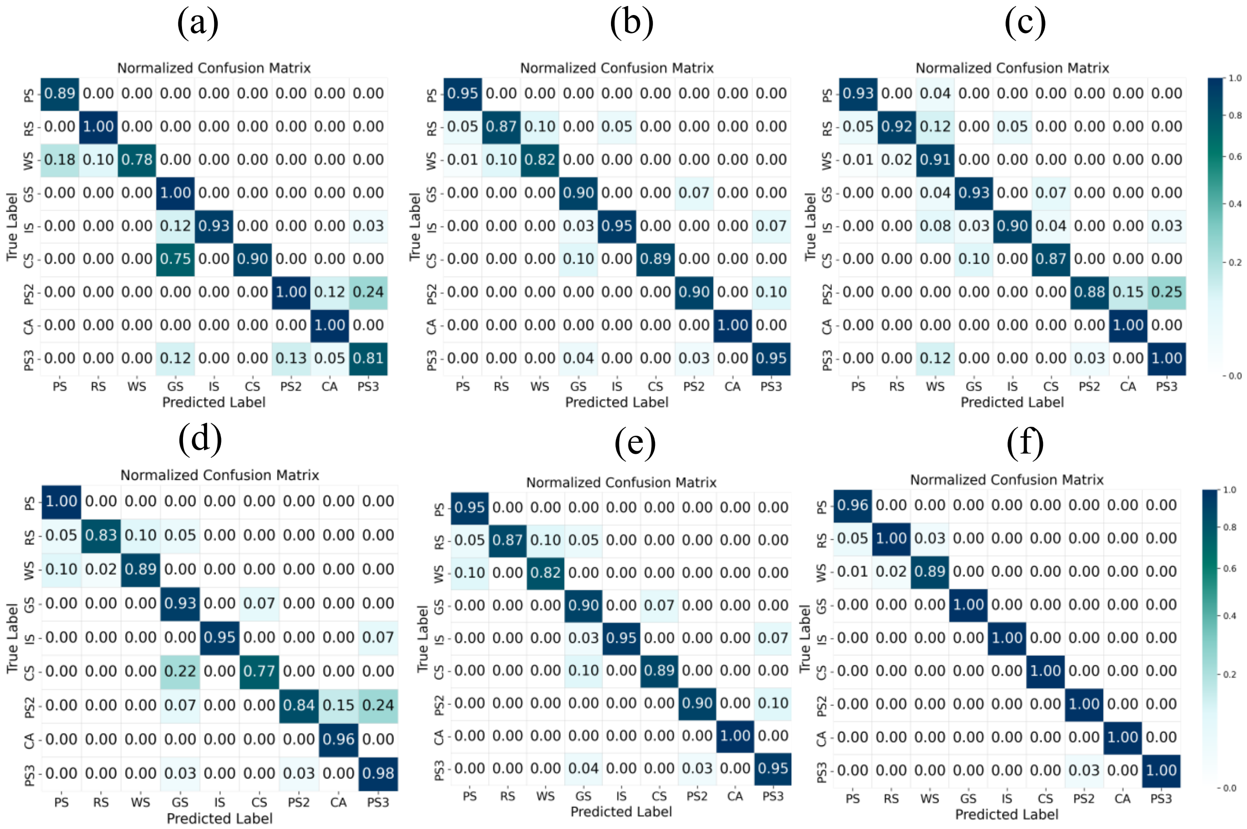

**Figure 11.** Confusion matrix of nine different classes of traffic signs. (a) SSD model. (b) Faster R-CNN model. (c) RepLKNet model. (d) DETR. (e) RepViT. (f) TTSNet (ours). Types of traffic signs: PS: prohibition signs; RS: regulatory signs; WS: warning signs; GS: guidance signs; IS: informational signs; CS: construction signs; PS2: parking signs; CA: cautionary signs; PS3: priority signs.

### *4.5. Confusion matrix analysis*

A confusion matrix reflects the degree to which an algorithm can make wrong predictions on similar classes. In this experiment, we have confirmed that the confusion matrix has been normalized. To ensure data accuracy, we conducted rigorous checks on numerical precision throughout the computation and rendering processes, preventing any deviation caused by loss of precision. These measures enable the confusion matrix to accurately reflect the degree to which the algorithm confuses and misclassifies similar classes, thereby improving the reliability of the analysis results. Nine different traffic sign classes were selected, namely prohibition signs (PS), regulatory signs (RS), warning signs (WS), guidance signs (GS), informational signs (IS), construction signs (CS), parking signs (PS2), cautionary signs (CA), and priority signs (PS3). These categories represent a diverse range of traffic signs, each requiring fine-grained recognition capabilities. Six methods are used for comparison. As Figure 11 shows, a lighter color in the confusion matrix denotes a higher percentage of correct predictions in the corresponding class. Ground truth labels are on the x-axis, while predicted labels are on the y-axis.

We observe that CNN-based methods [Figure 11(a)–(c)] are prone to confused predictions, particularly among visually similar classes such as cautionary signs (CA) and warning signs (WS) or prohibition signs (PS) and priority signs (PR3). In contrast, Transformer-based methods [Figure 11(d)–(f)] demonstrate better performance by leveraging their ability to model long-range dependencies and capture subtle distinctions in the visual characteristics of traffic signs. Among all methods, our proposed method [Figure 11(f)] exhibits the best performance, effectively distinguishing similar traffic signs. This success can be attributed to its ability to learn invariant features (e.g., color and shape consistency). It also captures subtle differences, such as symbol details or text variations, which helps minimize confusion among visually similar categories. Additionally, the T100K dataset does not suffer from class imbalance. The distribution of traffic signs in the dataset is relatively complete, ensuring that each category has sufficient representation for model training and evaluation. This balanced distribution allows for a fair comparison of classification performance across different methods. The confusion matrix in Figure 11 highlights which types of traffic signs are most prone to misclassification. Specifically, certain visually similar categories, such as cautionary signs (CA) and warning signs (WS) or prohibition signs (PS) and priority signs (PS3), exhibit higher confusion rates. This suggests that these classes share common visual elements that challenge recognition models. This balanced distribution allows for a fair comparison of classification performance across different methods.

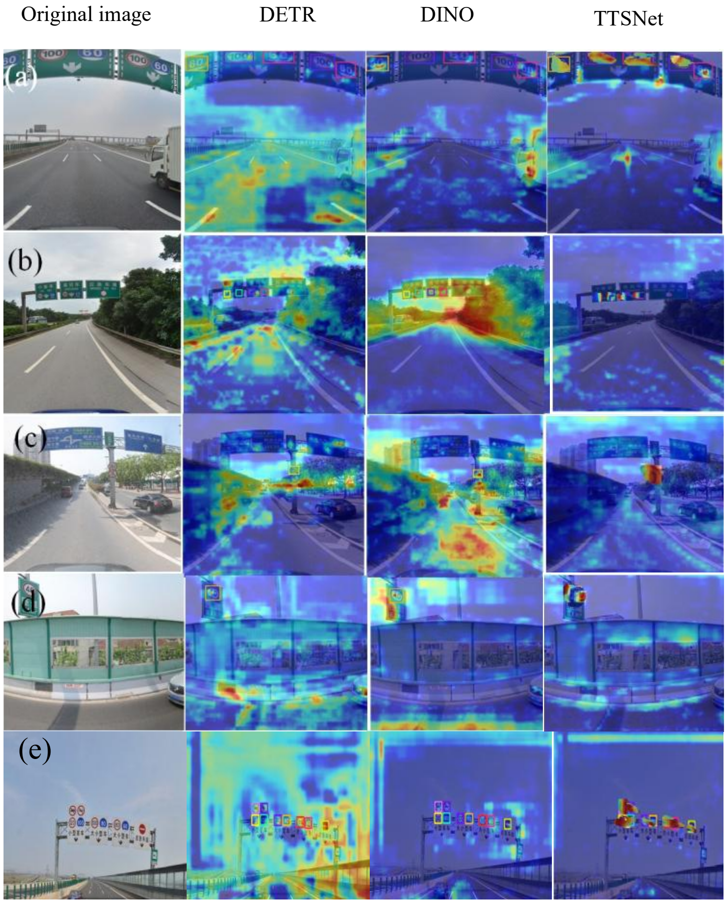

**Figure 12.** Traffic sign visualization heatmap.

*4.6. Visualization analysis*

Figure 12 illustrates five groups of traffic scene images [Figure 12(a)–(e)], with each group showing the original image on the left, followed by attention heatmaps generated by three different models: DETR, D-DETR, and our proposed TTSNet (from left to right). The attention maps demonstrate that TTSNet focuses more precisely on the traffic signs, exhibiting stronger target localization and more concentrated feature activation in the relevant regions.

In contrast, DETR and D-DETR produce more scattered attention, with significant focus on irrelevant areas such as vehicles, road surfaces, or vegetation. These distractions indicate weaker discrimination and context understanding in complex scenes. TTSNet not only accurately captures all traffic signs but also reveals the relationships between different signs, thanks to its enhanced feature interaction and hierarchical attention mechanism. This highlights TTSNet's superior capability in extracting and integrating task-relevant features, leading to more robust and reliable traffic sign recognition.

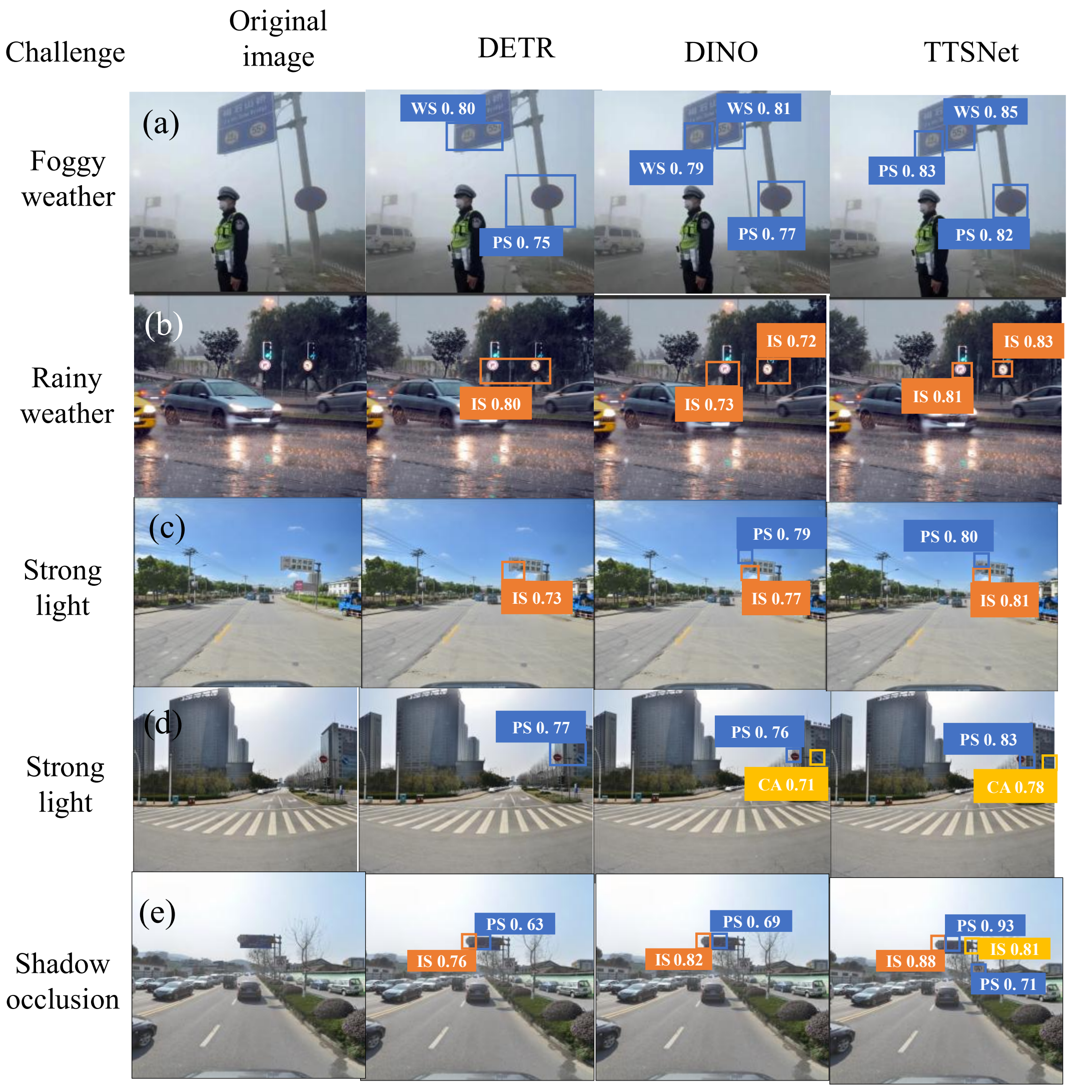

**Figure 13.** Visual comparison of different algorithms on the T100K dataset. Types of traffic signs: PS: prohibition signs; WS: warning signs; IS: informational signs; CA: cautionary signs.

Figure 13 showcases the detection results across five real-world traffic scenes under challenging conditions: (a) heavy fog, (b) rain and reflective night lighting, (c) bright daylight with distant signs, (d) complex urban intersections, and (e) dense traffic with occlusions. From left to right, each row displays the outputs from DETR, D-DETR, C-DETR, and our proposed TTSNet. Across all scenarios, TTSNet consistently detects more traffic signs with higher confidence scores. For example, in foggy and rainy scenes [Figure 13(a)–(b)], where visibility is significantly reduced, TTSNet is still able to accurately identify both warning and instruction signs, while other models either fail to detect them or misclassify unrelated objects as signs. In Figure 13(c), TTSNet successfully detects small, distant signs missed by the others, and in Figure 13(d), it correctly identifies multiple sign types even against complex urban backdrops. In Figure 13(e), which contains multiple vehicles and partially occluded signs, TTSNet achieves accurate localization and classification with confidence scores as high as 0.93, outperforming the other models that show either missed detections or poor localization. TTSNet's advantage lies in its ability to model long-range dependencies and contextual relationships between different signs, enabling it to handle small-scale, overlapping, or visually ambiguous signs with precision. These consistent improvements across a wide range of difficult conditions clearly demonstrate that TTSNet offers superior robustness, generalization, and multi-target discrimination capabilities compared to DETR, D-DETR, and C-DETR, making it highly suitable for complex real-world traffic environments.

Table 7 presents the quantitative evaluation of our model under different adverse weather conditions. For each weather scenario—fog, rain, backlight, and occlusion—we report the mAP@0.5, mAP@[0.5:0.95], and small-object mAP. The results demonstrate that while overall performance slightly decreases under more challenging conditions, the model maintains robust detection capabilities across all adverse-weather scenarios.

**Table 7.** mAP metrics for different weather scenarios.

| Weather | *mAP@*.5 (%) | *mAP@*.5:95 (%) | Small-object mAP |
|---|---|---|---|
| Fog | 92.3 | 78.5 | 70.2 |
| Rain | 90.1 | 76.8 | 68.9 |
| Backlight | 88.7 | 75.0 | 66.5 |
| Occlusion | 85.2 | 72.3 | 63.7 |

## 5. Conclusion

In this work, we propose a method for learning core features and uncovering long-term semantic relationships within internal features for traffic sign recognition. Additionally, we address two challenges: the complexity of backgrounds and the confusion caused by occlusions in images. We address these issues by introducing an efficient TTSNet model that leverages the ESIR module to extract core features, the DLFI module to capture long-term semantic dependencies, and the TTSM-Fusion module to merge the features. Extensive experiments show that TTSNet achieves state-of-the-art performance on the T100K (89.1%) and CTSDB (89.97%) datasets. The results on these two datasets demonstrate that TTSNet performs exceptionally in identifying key features for traffic sign recognition tasks. With ongoing advancements in technology, including the development of more efficient architectures and the integration of self-supervised and reinforcement learning methods, we expect that TTSNet's accuracy and speed will continue to improve. Furthermore, as datasets become more diverse and representative of global traffic signs, the model's generalization capabilities are likely to be enhanced. The application scope of TTSNet has the potential to extend beyond traditional road signs, possibly encompassing intelligent transportation systems, autonomous driving, and augmented reality in the future.

## Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

## Conflict of interest

All authors declare no conflicts of interest in this paper.

## Acknowledgements

This work was supported by the Science and Technology Talent Service Enterprise Program of Hubei Province under Grant 2025DJB050, the University Fund of Wuhan Textile University (Project No. k24082), and in part by Jiangxi Provincial Natural Science Foundation under Grant 20252BAC220007, Grant 20252BAC240201, Grant 20242BAB2S107, and Grant 20232BAB212026; in part by the National Natural Science Foundation of Hubei Province under Grant 2025AFD621; in part by Shenzhen Science and Technology Program under Grant JCYJ20250604185710014 and Grant JCYJ20230807152900001; in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2025A1515010266; in part by the Fundamental Research Funds for the Central Universities under Grant CCNU25ai012 and Grant CCNU25XJ003; and in part by the Science and Technology Research Program of Hubei Provincial Department of Education under Grant B2024301 and Grant 2025DHWL002.

## References

1. K. Wang, L. J. Zheng, B. Lin, Demand-side incentives, competition, and firms' innovative activities: Evidence from automobile industry in China, *Energy Econ.*, **132** (2024), 107426. https://doi.org/10.1016/j.eneco.2024.107426
2. H. Liu, H. Nie, Z. Zhang, Y.-F. Li, Anisotropic angle distribution learning for head pose estimation and attention understanding in human–computer interaction, *Neurocomputing*, **433** (2021), 310–322. https://doi.org/10.1016/j.neucom.2020.09.068
3. L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, H. Li, End-to-end autonomous driving: Challenges and frontiers, *IEEE Trans. Pattern Anal. Mach. Intell.*, (2024). https://doi.org/10.1109/TPAMI.2024.3435937
4. Y. Xu, T. Fu, H. Yang, C. Lee, Dynamic video segmentation network, in: *Proc. IEEE Conf.*

*Comput. Vis. Pattern Recognit.*, 2018, 6556–6565. https://doi.org/10.1109/CVPR.2018.00686

5. H. Liu, S. Fang, Z. Zhang, D. Li, K. Lin, J. Wang, MFDNet: Collaborative poses perception and Matrix Fisher distribution for head pose estimation, *IEEE Trans. Multimedia*, **24** (2022), 2449–2460. https://doi.org/10.1109/TMM.2021.3081873
6. W. Rawat, Z. Wang, Deep convolutional neural networks for image classification: A comprehensive review, *Neural Comput.*, **29** (2017), 2352–2449. https://doi.org/10.1162/neco_a_00990
7. D. Wang, K. Mao, Learning semantic text features for web text aided image classification, *IEEE Trans. Multimedia*, **21** (2019), 2985–2996. https://doi.org/10.1109/TMM.2019.2920620
8. Y. Zhong, Y. Wei, Y. Liang, X. Liu, R. Ji, Y. Cang, A comparative study of generative adversarial networks for image recognition algorithms based on deep learning and traditional methods, arXiv:2408.03568, 2024. https://doi.org/10.1109/ICPICS62053.2024.10797049
9. H. Liu, Y. Song, T. Liu, J. Ju, J. Tang, UGENet: Learning discriminative embeddings for unconstrained gaze estimation network via self-attention mechanism in human–computer interaction, in: *Proc. Int. Conf. Comput. Inf. Big Data Appl.*, Wuhan, China, 2024. https://doi.org/10.1145/3671151.3671152
10. W. Ge, X. Lin, Y. Yu, Weakly supervised complementary parts models for fine-grained image classification from the bottom up, in: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, 3034–3043.
11. C. Liu, H. Xie, Z. J. Zha, L. Ma, L. Yu, Y. Zhang, Filtration and distillation: Enhancing region attention for fine-grained visual categorization, in: *Proc. AAAI Conf. Artif. Intell.*, **34** (2020), 11555–11562. https://doi.org/10.1609/aaai.v34i07.6822
12. Q. Fan, W. Zhuo, C. K. Tang, Y. W. Tai, Few-shot object detection with attention-RPN and multi-relation detector, in: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* 2020, 4013–4022. https://doi.org/10.1109/CVPR42600.2020.00407
13. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556, 2014.
14. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, 770–778. https://doi.org/10.1109/CVPR.2016.90
15. G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, 4700–4708. https://doi.org/10.1109/CVPR.2017.243
16. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., Going deeper with convolutions, in: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, 1–9. https://doi.org/10.1109/CVPR.2015.7298594
17. W. Luo, X. Yang, X. Mo, Y. Lu, L. S. Davis, J. Li, et al., Cross-X learning for fine-grained visual categorization, in: *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, 8242–8251. https://doi.org/10.1109/ICCV.2019.00833
18. P. Zhuang, Y. Wang, Y. Qiao, Learning attentive pairwise interaction for fine-grained classification, in: *Proc. AAAI Conf. Artif. Intell.*, 2020, 13130–13137. https://doi.org/10.1609/aaai.v34i07.7016

19. R. Du, J. Xie, Z. Ma, D. Chang, Y. Z. Song, J. Guo, Progressive learning of category-consistent multi-granularity features for fine-grained visual classification, *IEEE Trans. Pattern Anal. Mach. Intell.*, **44** (2022), 9521–9535. https://doi.org/10.1109/TPAMI.2021.3126668
20. S. Pouyanfar, S. Sadiq, Y. Yan, H. M. Tian, Y. D. Tao, M. Presa, et al., A survey on deep learning: Algorithms, techniques, and applications, *ACM Comput. Surv.*, **51** (2018), 1–36. https://doi.org/10.1145/3234150
21. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., Attention is all you need, *Adv. Neural Inf. Process. Syst.*, **30** (2017).
22. J. He, J. Chen, S. Liu, A. Kortylewski, C. Yang, Y. Bai, et al., TransFG: A transformer architecture for fine-grained recognition, *Proc. AAAI Conf. Artif. Intell.*, 2022, 852–860. https://doi.org/10.1609/aaai.v36i1.19967
23. H. Liu, S. Zeng, L. Deng, T. Liu, X. Liu, Z. Zhang, et al., HPCTrans: Heterogeneous plumage cues-aware texton correlation representation for FBIC via transformers, *IEEE Trans. Circuits Syst. Video Technol.*, (2025), https://doi.org/10.1109/TCSVT.2025.3601504
24. S. Branson, G. Van Horn, S. Belongie, P. Perona, Bird species categorization using pose normalized deep convolutional nets, arXiv:1406.2952, 2014. https://doi.org/10.5244/C.28.87
25. Z. Zhao, P. Zheng, S. Xu, X. Wu, Object detection with deep learning: A review, *IEEE Trans. Neural Netw. Learn. Syst.*, **30** (2019), 3212–3232. https://doi.org/10.1109/TNNLS.2018.2876865
26. M. H. Guo, T. X. Xu, J. J. Liu, Z. N. Liu, P. T. Jiang, T. J. Mu, et al., Attention mechanisms in computer vision: A survey, *Comput. Vis. Media*, **8** (2022), 331–368. https://doi.org/10.1007/s41095-022-0271-y
27. T. Liu, H. Liu, B. Yang, Z. Zhang, LDCNet: Limb direction cues-aware network for flexible HPE in industrial behavioral biometrics systems, *IEEE Trans. Ind. Inf.*, **20** (2024), 8068–8078. https://doi.org/10.1109/TII.2023.3266366
28. N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, aS. Zagoruyko, End-to-end object detection with transformers, in: *Proc. Eur. Conf. Comput. Vis.*, 2020, 213–229. https://doi.org/10.1007/978-3-030-58452-8_13
29. H. Liu, T. Liu, Y. Chen, Z. Zhang, Y. Li, EHPE: Skeleton cues-based Gaussian coordinate encoding for efficient human pose estimation, *IEEE Trans. Multimedia*, **26** (2024), 8464–8475. https://doi.org/10.1109/TMM.2022.3197364
30. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. H. Zhai, T. Unterthiner, et al., An image is worth 16×16 words: Transformers for image recognition at scale, in: *Proc. Int. Conf. Learn. Represent.*, 2021.
31. C. R. Chen, Q. Fan, R. Panda, CrossViT: Cross-attention multiscale vision transformer for image classification, in: *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, 347–356. https://doi.org/10.1109/ICCV48922.2021.00041
32. Z. Liu, Y. T. Lin, Y. Cao, H. Hu, Y. X. Wei, Z. Zhang, et al., Swin transformer: Hierarchical vision transformer using shifted windows, in: *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, 9992–10002. https://doi.org/10.1109/ICCV48922.2021.00986
33. J. Fang, L. Xie, X. Wang, X. Zhang, W. Liu, Q. Tian, MSG transformer: Exchanging local spatial information by manipulating messenger tokens, in: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern*

*Recognit.*, 2022, 12053–12062. https://doi.org/10.1109/CVPR52688.2022.01175

34. H. Liu, C. Zhang, Y. Deng, B. Xie, T. Liu, Z. Zhang, et al., TransIFC: Invariant cues-aware feature concentration learning for efficient fine-grained bird image classification, *IEEE Transact. Multimedia*, **27** (2025), 1677–1690. https://doi.org/10.1109/TMM.2023.3238548
35. Y. Li, S. K. Zhang, Z. C. Wang, S. Yang, W. K. Yang, S. T. Xia, et al., TokenPose: Learning keypoint tokens for human pose estimation, in: *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, 11293–11302. https://doi.org/10.1109/ICCV48922.2021.01112
36. H. Liu, T. Liu, Z. Zhang, A. K. Sangaiah, B. Yang, Y. Li, ARHPE: Asymmetric relation-aware representation learning for head pose estimation in industrial human–computer interaction, *IEEE Trans. Ind. Inf.*, **18** (2022), 7107–7117. https://doi.org/10.1109/TII.2022.3143605
37. Y. Sun, S. Ma, S. Y. Sun, P Liu, L. Zhang, J. Ouyang, et al., Partial discharge pattern recognition of transformers based on MobileNets convolutional neural network, *Appl. Sci.*, **11** (2021), 6984. https://doi.org/10.3390/app11156984
38. X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable DETR: Deformable transformers for end-to-end object detection, arXiv:2010.04159, 2020.
39. W. Han, N. He, X. Wang, F. Sun, S. Liu, IDPD: Improved deformable-DETR for crowd pedestrian detection, *Signal Image Video Process.*, **18** (2024), 2243–2253. https://doi.org/10.1007/s11760-023-02896-2
40. D. Meng, X. Chen, Z. Fan, G. Zeng, H. Li, Y. Yuan, et al., Conditional DETR for fast training convergence, in: *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, 3651–3660. https://doi.org/10.1109/ICCV48922.2021.00363
41. H. Liu, Q. Chen, Z. Liu, T. Liu, L. Zhao, Z. Zhang, et al., SkeFormer: Skeletal cues-aware bone point relationship learning for efficient FBIC via transformers, *IEEE Trans. Multimedia*, (2025), https://doi.org/10.1109/TMM.2025.3603431
42. X. Hou, M. Liu, S. Zhang, P. Wei, B. Chen, Salience DETR: Enhancing detection transformer with hierarchical salience filtering refinement, in: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, 17574–17583. https://doi.org/10.1109/CVPR52733.2024.01664
43. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE Trans. Pattern Anal. Mach. Intell.*, **39** (2016), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
44. L. Zhang, C. Xing, F. Gao, T.-S. Li, Y.-Q. Tan, Using DSR and MSCR tests to characterize high temperature performance of different rubber modified asphalt, *Constr. Build. Mater.*, **127** (2016), 466–474. https://doi.org/10.1016/j.conbuildmat.2016.10.010
45. R. Wang, Large kernel convolutional neural networks for action recognition based on RepLKNet, in: *Proc. Int. Conf. Image Process. Comput. Vis. Mach. Learn.*, 2023, 103–107. https://doi.org/10.1109/ICICML60161.2023.10424920
46. H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, et al., DINO: DETR with improved denoising anchor boxes for end-to-end object detection, arXiv:2203.03605, 2022.
47. L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. H. Jiang, et al., Tokens-to-token ViT: Training vision transformers from scratch on ImageNet, in: *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, 558–567. https://doi.org/10.1109/ICCV48922.2021.00060