IET Intelligent Transport Systems

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

Attention-Guided Lightweight CNN-Transformer Fusion for Real-Time Traffic Sign Recognition in Adverse Environments: HACTNet

Mandeep Singh Devgan[1] | Gurvinder Singh[2] | Purushottam Sharma[3] | Tajinder Kumar[4] | Xiaochun Cheng[5] | Deepak Ahlawat[2]

[1]Department of Computer Science Engineering, University Institute of Engineering, Chandigarh University, Mohali, Gharuan, Punjab, India | [2]Department of Computer Science and Engineering, Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India | [3]School of Computer Science and Engineering, Galgotias University, Greater Noida, India | [4]Department of Computer Science and Engineering, Jai Parkash Mukand Lal Innovative Engineering and Technology Institute, Radur, India | [5]Computer Science Department, Bay Campus Fabian Way, Swansea University, Swansea, UK

Correspondence: Purushottam Sharma (Puru.mit2002@gmail.com) | Xiaochun Cheng (xiaochun.cheng@swansea.ac.uk)

Received: 2 September 2025 | **Revised:** 30 November 2025 | **Accepted:** 6 February 2026

ABSTRACT

Autonomous driving is also impossible without traffic sign recognition (TSR; also known as traffic sign-on-road), which limits its reliability to domain changes, unfavourable weather, obstruction and hardware capacity. This paper proposes HACTNet, a low-complexity CNN-Transformer hybrid model that pushes the state-of-art in TSR by making a noteworthy set of contributions including (i) efficient convaps to model parts of the image, (ii) transformer encoder to capture the global context and (iii) an attention-based fusion block to dynamically combine the two complementary sets of features. This synergy facilitates strong recognition in presence of blur and occlusion and in varying illumination. In addition to accuracy, HACTNet achieves high robustness (52.8%) against strong PGD adversarial attacks (8/255), but is still efficient (7.9 M parameters and 22.1 FPS) on the NVIDIA Jetson Nano. Moreover, the comparative analysis between the hybrid models (EATFormer, local-ViT) and HACTNet proves that HACTNet has a better accuracy-efficiency ratio. The extraordinary capability to counteract adverse weather conditions, fog, night, rain, snow etc., which is proven by the extensive testing of the real-world ACDC adverse conditions data set, supports the viability of the proposed solutions in the real world. It is plug and play modularity with on-going learning via elastic weight consolidation (3.3% less forgetting) and unsupervised domain adaptation via MMD loss (5.3% better on TT100K with no labels). Moreover, INT8 quantization with quantization-aware training (QAT) incurs little accuracy loss (less than 0.5 percent) and much lower energy (0.27 J/sample) usage, which forms an edge deployment preparedness. Additionally, when adjusting to new traffic signs over time, the model shows compatibility with continuous learning, achieving a low forgetting rate (3.3%), highlighting its practical viability for long-term autonomous deployment. Overall, HACTNet produces a versatile and expandable solution for next-generation intelligent transportation systems by striking a balance between accuracy, robustness and efficiency.

1 | Introduction

Traffic sign recognition (TSR) is an important part of intelligent transportation systems (ITS) and advanced driver-assistance systems (ADAS) with the ability to detect and recognize regulatory, warning and informational traffic signs in real-time to support autonomous navigation [1–3].

Early TSR techniques used manually defined characteristics (e.g. shape/colour segmentation) combined with more conventional classifiers such as SVM [4], KNN or decision trees [5–7]. Although these methods worked well in controlled environments, they were unable to generalise to real-world conditions (e.g. occlusions, change in lighting, weather perturbations) [8]. Due to the advent of deep learning, especially CNNs, TSR was redefined as extracting features is now automatic, reaching an accuracy of over 98 percent in benchmark tasks, such as GTSRB and BTSC [9–14]. But CNNs suffer due to the limited local receptive fields, which inhibit modeling long-range dependencies. This was solved in vision transformers (ViTs) through self-attention but showed poor performance on small or imbalanced datasets because of low inductive bias [15–18]. New hybrid CNN-Transformer models (e.g. LVT [19], EATFormer [20], EfficientNet-ViT [21]) leverage both local feature extraction and global context modeling to achieve greater accuracy and efficiency to be deployed on the edge. Additional robustness is achieved with data augmentation, that is occlusions, adversarial perturbations and photometric distortions [22–24]. The most prominent challenge is cross-dataset generalization, whereby models trained on GTSRB can fail on BelgiumTS or TT100K or region-specific datasets since the design or environmental variances are not similar [25, 26]. Moreover, the latency, memory and power requirements in the real-world imply optimization on the edge devices (e.g. NVIDIA Jetson, Raspberry Pi) [27, 28].

1.1 | What the Work Has to Offer-Work Contributions

In order to solve them, this paper proposes a lightweight hybrid CNN-transformer-based TSR system, the main contributions of which are as follows:

1. New hybrid architecture: We propose a CNN infrastructure that integrates with attention-based transform sections to make use of both local and global feature-extraction capability.
2. Realistic data augmentation: The augmentation pipeline with occlusion, blur and distortion, adversarial examples is built to enhance model performance to be generalizable.
3. Cross-dataset testing: Model is trained on GTSRB and tested on BelgiumTS and the regional traffic sign dataset in order to test its robustness.
4. Ablation studies: Experiments on the size of patches, heads of attention, dropout rates and CNN depth are detailed, which maximize the performance of the architecture.
5. Edge deployment metrics: Deployment feasibility is presented by reporting the real-time inference speed, memory footprint and energy consumption on Jetson Nano.
6. Statistical significance: The data is tested as valid through 5-fold cross-validation and paired t-tests of all the performance gains.

Our system can produce competitive results (<23.1 FPS) on several benchmarks (<99.6% accuracy on GTSRB) and be integrated into real-world autonomous systems due to the number of advances described in this section aimed to ensure a high inference speed, small inference time and a small model size.

1.2 | The Key Innovations and Benefits of Proposed HACTNet

Compared with the existing hybrid CNN-transformer solutions such as Farzipour et al. [29] that employs local vision transformers with pre-determined locality modules or Mingwin et al. [30] that has a heavy pyramid EATFormer structure, our proposed HACTNet presents an attention-guided feature fusion component that fully adapts local and global representations. This dynamic cross-attention module enables the network to give greater or lesser emphasis to spatial and semantic features depending on context better robustness to occlusion, blur and changes in lighting. Additionally, [29] and [30] concentrate on accuracy and generalization, whereas HACTNet adds real-time inference and edge deployment as the main emphasized issues, where speed, power consumption and memory consumption are benchmarked on the Jetson Nano which is underrepresented in prior studies.

Finally, the model is rigorously validated not only on standard benchmarks but also against state-of-the-art hybrids and in real-world adverse weather scenarios, providing comprehensive evidence of its practical advantages.

2 | Related Work

The development of efficient and viable models in the form of traffic sign recognition has been increasingly involved over the time frame 2021-mid 2025 due to the necessity of effective models that can be applied practically in the real world environment and in the edge environments. This section points out strong and weak aspects of the remarkable pieces on architectural novelty, enhancement of the performance benchmark, use of datasets, robustness and efficiency of inference.

2.1 | Light CNNs Real-Time and Edge Deployment

Bangquan et al. [31] suggest ENet and EmdNet built on depthwise separable convolutions and operate in real-time with a high accuracy rate. They obtain ~98.40\percent, GTSRB and 92.10\percent, BelgiumTS but the number of parameters required to perform inference is small [31]. Likewise, the minimalist variant of LeNet 5 proposed by Zaibi et al. [21] has only 0.38 M parameters but yields 99.84 % on GTSRB and 98.37 % on BTSC, which qualifies it to work as an extremely efficient implementation of online TSR [31]. They both outline efficiency of computation without much compromise on accuracy.

Madan et al. (2022) propose a branching CNN that combines HOG and SURF features using two parallel convolutional paths and achieve 98.48 % accuracy on GTSRB-only slightly better than standard CNNs but at a slight cost [32]. The method emphasis is the combination of hand crafted and actually learned features in constrained resource situations.

2.2 | The Methods of Data Augmentation and Robustness-Focused CNNs

The explained CNN framework by Khan et al. (2023) incorporates guided filtering and data augmentation into a network that is used to offset the problem of occlusions and motion blur and succeeds in surpassing 99.6 % F1 score on TT100K which is higher than that of ResNet 34 and AlexNet baseline models [33]. That study reaffirms that preprocessing and augmentation influences the domain robustness. Toshniwal et al. (2024) describe an improved CNN pipeline performance of ∼96 per cent accuracy on GTSRB through localization refinement in addition to tuning the classifier to improve performance in the presence of real-world variability [34]. Being not as precise as the heavier models do, their emphasis on robustness and adapting to the domain is striking.

2.3 | Hybrid Transformer Models and Vision Transformer (ViT)

Farzipour et al. (2023) introduce a local-vision-transformer hybrid system, where CNN-based local features extraction is integrated with Transformer blocks with the locality module. On GTSRB and a Persian traffic sign dataset, this model obtains 99.66 99.8 99.8 % better than both purer CNNs and ViTs and at inference speed appropriate to edge devices [29].

Mingwin et al. (2024) present EATFormer, a pyramid Transformer with module design being trained on an evolutionary algorithm and a right to remain self-attention deformable multi-head attention. BAfinTrained on GTSRB and BelgiumTS, EATFormer surpasses the baselines (PVT, TNT, LNL and EfficientNet variants) in BelgiumTS accuracy of 92.16 % with only ∼9.6 M parameters compared to ResNet or EfficientNet having ∼40 M [30]. Their strategy is very generalizing in terms of datasets using less parameters. Mirzapour Kaleybar et al. (2023) provide an efficient vision transformer to detect traffic signs on the basis of compact but competitive transformer variants under poor lighting conditions and moving objects single-image detection [35]. Detection-oriented, their work supports the feasibility of lightweight transformer models in TSR, too.

Although Farzipour et al. [29] and Mingwin et al. [30] presented efficient CNN-based boom hybrids, our HACTNet is different in two major aspects:

1. **Fusion mechanism**: Our approach is an attention-guided cross-fusion block that calculates a context-sensitive weight to mix CNN cues and Transformer [36, 37] cues in contrast to [29] which applies locality-aware modules or [30] which employs a pyramid-type fusion.
2. **Edge-deployment design**: Indeed, HACTNet is directly targeting low-power edge devices, which, with just 7.9 M parameters, reaches 22.1 FPS on Jetson Nano, compared with ∼9.6 M parameters in [30], which does not report on its real-time performance on constraint hardware.

The differences illustrate the specific role of HACTNet as a path to scalable, generalizable and deployable TSR models under the variability of the real world.

2.4 | CLIP Based Cross Regional Generalization

Zhao et al. (2024) advance TSCLIP: fine tuning CLIP to traffic sign [38, 39] recognition on various regions through prompt engineering and dynamic weighting ensembling. They select a multi regional benchmark of ten sources and report SOTA performance in cross regional generalization-showing the zero shot expansion of CLIP to TSR [40]. TSCLIP highlights the promising prospect of language vision models in the processing of regionally variant sign set.

2.5 | Comparative Overview

An analogical and complementary area of study is on end-to-end based models of detection such as the enhanced YOLOv5 networks [19, 20], which are notable in real-time, multi-scale localization of traffic signs under complex circumstances. Although these detectors are essential in the detection of areas of interest, their architectural optimization is mostly focused on spatial reasoning and analysis of multiple objects in a complete image frame. The next step of fine-grained recognition, which is to be able to spot a possibly distorted, covered-up or darkened sign correctly is also a separate and problematic issue.

This stage of recognition is specifically the target of our own recognition, HACTNet. We assume that the overall system reliability requires a specific model, which is not constructed with localization in mind but focused on the highest robustness in classification. As such, the nature of our contributions is quite distinct: we do not change a detection backbone, rather we present a new hybrid recognition backbone the main innovation of which is a lightweight attention-guided fusion mechanism. The module is specifically meant to overcome ambiguity prevalent in case of adverse conditions through dynamic combination of features. Moreover, we offer an end-to-end edge-deployment analysis such as energy usage and latency breakdown on a Jetson Nano, that is generally essential to the real-world integration but is usually not considered by detection-model evaluations. Therefore, HACTNet is a high-confidence classification module, which can be easily plugged into the bottom of any contemporary detector to increase the stability of the overall traffic sign perception pipeline.

Table 1 below contrasts contemporaneous TSR techniques in terms of architecture, datasets and accuracy, efficiency and deployment priority, as well as noting major limitations, prioritizing trade-offs out of that performance, robustness and running-in real time.

Recently, YOLO-TS [41] was proposed as a more powerful real-time traffic sign detector, which uses optimized receptive fields and anchor-free fusion to enhance the detection accuracy of signs of different sizes. Although YOLO-TS has been shown as superior in the localization and multi-object detection of objects in the entire image frame, its architectural design is different to that of HACTNet, which has a specific classification goal. HACTNet is not intended to be localized but to be recognized with high accuracy, especially in the presence of occlusion, blur and domain shifts and is therefore a complementary component that can be used after detectors such as YOLO-TS.

TABLE 1 | Comparative overview.

Method	Architecture	Datasets	Accuracy (%)	Params/ efficiency	Deployment focus	Limitations of existing methods
Zaibi et al. [21]	Tiny LeNet-5 (highly compressed convolutional network with minimal layers)	GTSRB, BTSC	99.84, 98.37	~0.38 M	Real-time, edge	Extremely lightweight but may underperform on complex, high-variance datasets due to limited feature capacity; lacks robustness to severe occlusions or domain shifts.
Bangquan et al. [31]	ENet + EmdNet (depthwise separable convolutions + multi-scale features)	GTSRB, BelgiumTS	~98.4, 92.1	Low DSP	Embedded systems	Optimized for low-power inference but experiences noticeable accuracy drop in cross-dataset tests (e.g. BelgiumTS); limited capacity for fine-grained context modeling.
Madan et al. [32]	Branching CNN + HOG/SURF handcrafted feature fusion	GTSRB	98.48	Moderate	Lightweight fusion	Relies partly on handcrafted features, reducing adaptability to unseen patterns; lacks global attention mechanisms, limiting performance under background clutter.
Khan et al. [33]	Explainable CNN + guided filter preprocessing	TT100K	~99.6 (F1-score)	Medium	Robustness in adverse conditions	Strong under noise/blur but computationally heavier due to preprocessing; may not meet strict latency constraints on ultra-low-power devices.
Toshniwal et al. [34]	Optimized CNN with localization refinement	GTSRB	~96	Standard CNN	Domain adaptation	Lower accuracy compared to hybrid or transformer-based models; struggles with high intra-class variation and complex occlusions.
Farzipour et al. [29]	CNN + local vision transformer hybrid with locality modules	GTSRB, PersianTS	99.66, 99.80	Compact hybrid	Accuracy + speed	Locality module improves efficiency but may miss long-range dependencies across the entire image; less optimized for extreme edge constraints.
Mingwin et al. [30]	Pyramid EATFormer (hierarchical transformer with deformable attention)	GTSRB, BelgiumTS	~92 (BelgiumTS)	~9.6 M	Generalization + efficiency	Good generalization but relatively high parameter count for embedded deployment; pyramid design adds complexity in training and tuning.

(Continues)

TABLE 1 | (Continued)

Method	Architecture	Datasets	Accuracy (%)	Params/ efficiency	Deployment focus	Limitations of existing methods
Mirzapour Kaleybar et al. [35]	Efficient Vision Transformer	GTSRB	Not specified	Lightweight ViT	Detection under variability	Prioritizes detection over classification; accuracy and robustness in fine-grained classification tasks not fully reported.
Zhao et al. [40]	CLIP-based fine-tuning with multi-regional traffic sign datasets	Multi-regional	SOTA cross-regional	Large pre-trained	Domain general recognition	Relies on large vision-language models with high computational cost; impractical for real-time edge deployment without distillation or compression.

Another simultaneous research impetus concerns the challenges of low-light conditions that are extremely demanding to the operation of conventional vision models. This issue is directly addressed by the recently introduced YOLO-LLTS [42], which combines a prior-guided low-light enhancement unit directly into a real-time-detection pipeline, and then a multi-branch feature interaction network. It is a dedicated design, and is quite suitable to the particular issue of low-light detecting and recognizing, maximizing the overall process of raw sensor data [43] to final prediction in poor lighting. Contrarily, HACTNet is designed to be a general-purpose recognition backbone which reaches robustness over a range of adverse scenarios (blur, occlusion, fog, rain, snow and night) due to its dynamic feature fusion, instead of preprocessing or condition-specific modules. This inherent disparity in design philosophy namely specialized enhancer-detector over a widely robust classifier necessarily renders YOLO-LLTS an interesting solution to specific low-light applications, whereas HACTNet is designed to be all-weather viable.

2.6 | Generalization and Lightweight Design in Broader Deep Learning Contexts

The issues of robust generalization and computational efficiency are not specific to TSR, but the main concern of using deep learning in real-world and resource-constrained systems. New developments in other areas can be useful in giving architectural clues. As an example, in wireless communications, the problem of modulation schemes in noisy, variable conditions is reflected in the adversarial and environmental problem of TSR. Here lightweight hybrid architectures have performed well [44]; MobileViT [45, 46] uses convolutional spatial inductive biases with the global receptive field of transformers to perform efficient image classification, which is a design philosophy that has a direct connection to developing compact vision models. Continuing to take advantage of this synergy, MobileRaT [47] proposes a dynamically fusing radio transformer which ad hoc fuses multi-scale features with automatic modulation classification in the drones, which serves as the inspiration behind our attention-guided fusion module. The systematic classification of the history of deep learning models to achieve improved generalization to channel distortions and noise made by surveys in the field, like [48], makes clear the importance of universal architectures resistant to domain shift.

On the same note, the advances made on vision work never cease to enhance our knowledge on feature representation. The core issue of object detection by transformers is re-thinking the multi-scale feature hierarchy in the detection transformer (DETR) [49] to process features in different scales successfully. The development of it shows the essential balance between the local accuracy and the global context, which is directly related to TSR when indicators are represented in different sizes and resolutions. Some of the architectural concepts discussed in the papers such as lightweight hybrid design, dynamic integration of multi-scales features, and robustness to input degradation give a wider perspective to the development of HACTNet. The context of our work lies within this cross-domain research thrust and employs and focuses these generalized concepts to the special, safety-critical problem of traffic sign recognition in an adverse environment.

3 | Gap and Opportunities

Despite the considerable advances in the field of traffic sign recognition (TSR), major drawbacks still exist in cross-dataset generalization, and scalability at architecture level in addition to deployment inefficiency. Lightweight CNNs including ENet and EmdNet [31] perform more lightweight computation and memory at the edge of inference but commonly lack generalizability to new domains or harsh conditions.

Special purpose attention can be enhanced by branching architectures [32] which promote multi-path feature extraction in the feature network, and explainable deep networks [33], which promote interpretability by means of saliency maps or attention mechanisms. With that said, however, both rely extensively on convolutional priors thus complicating the modeling of long-range dependencies and flexibility to a wide variety of heterogeneous traffic conditions.

Architectures based on transformers, such as those with global self-attention, are highly effective at modeling spatially distant context, but have quadratic token complexity, high parameter counts and hardware-unfriendly operations which make them ineffective in latency-sensitive TSR. More modern hybrids, like the CNN-transformer models of Farzipour et al. [29] and Mingwin et al. [30] enable these with hierarchical fusion or parallel pathway designs, which maintain low complexity and increase robustness to viewpoint change, occlusion and domain shift. TSCLIP [40] adopts a different approach, which leveraged CLIP vision-language embeddings to model semantic similarity on a region-wise level to enable wide zero-shot adaptation.

Nevertheless, the use of large pretrained model limits real-time and embedded applicability with large inference cost (e.g. ViT-B/32, RN50 × 64).

There are these gaps which lead to the opportunity of modular cost-effective hybrids combining spatial locality with global context modeling with maintaining the domain generalization and the real-time efficiency particularly in the illumination variance, motion blur and noise induced by the weather.

In this direction, HACTNet is proposed, a lightweight and transformer-based hybrid that integrates local localization ability of CNNs [50, 51] and global context representation of transformers through an attention-controllable dynamic fusion block to result in strong, real-time TSR capability under adversarial conditions. For clarity, Table 1 provides an easy-to-follow summary of this discussion.

4 | Proposed Method

In order to overcome the drawbacks of the traditional CNN-based traffic sign recognition (TSR) systems to some extent in terms of occlusions, environmental variability and spatial deformations, we submit a hybrid attention-enhanced CNN-transformer network (HACTNet). This architecture is synergistic in that it uses the local feature extraction characteristic of convolutional layers in combination with global context modeling enabled by self-attention implemented by transformer. Our approach does promise to improve on both recognition accuracy and robustness under real-world traffic conditions, with computation cost to remain affordable and endurable even at a real-time environment. Workflow of the proposed model is illustrated in Figure 1 below.

The Figure 1 explains the proposed method, which begins with pre-processing the raw image and extracting complementary spatial features using a CNN backbone alongside contextual representations from a transformer encoder. These extracted multi-scale features are fused through an attention-guided mechanism to create a unified and highly discriminative feature representation for classification. The classification output is evaluated using standard performance metrics to determine the overall effectiveness of the hybrid model.

4.1 | Architectural Overview

The proposed HACTNet architecture will be able to identify traffic signs with power due to the collective advantages of convolutional and transformer-based models. It includes five main modules, each of which are optimized to a certain step of the recognition pipeline:

1. **Preprocessing and data augmentation layer**: This phase normalizes input images and down-sizes them as well as performs synthetic forms of augmentation (e.g. change of brightness, rotates, noise) to enhance generalization to different lighting and environmental states.
2. **Convolutional feature extraction backbone (CNN block)**: A depthwise separable convolutional network with residual connections of light-weight builds low and mid-level features, including edges, textures and local patterns. This block will guarantee effective computation that will be applicable in real-time.
3. **Transformer encoder for global contextual modelling**: Image patches are tokenized and sent through a transformer encoder using multi-head self-attention and feed-forward layers in order to learn global dependencies and long-range interactions between features. This module improves the spatial awareness as well as contextual reasoning which is essential when faced with occlusion or clutter.
4. **Attention-guided feature fusion module**: There is dynamic combination of both CNN and transformer streams features with attention-based weighting mechanisms. This combination gives rise to a good balance between local details and global understanding of the scene, which produce strong and discriminative representations.
5. **Classification head with softmax layer**: The fused feature vector is converted into fully connected layers and a final softmax activation to give a prediction of traffic sign classes. This head is lightweight and it can be deployed on edge devices with resource limitations. In general, HACTNet combines local accuracy and global context to come up with effective, robust and accurate traffic signs recognition. Figure 2 indicates these modules.

Figure 3, ahead shows the proposed hybrid deep learning model to recognize traffic signs (TSR), aiming at leveraging the advantages of both convolutional neural networks (CNNs) and vision transformers (ViTs) in terms of accuracy and the ability to run efficiently in real-world variability.

The pipeline begins by loading a raw image of RGB traffic scene into the system, after which a series of data augmentation and preprocessing routines are applied in order to shuffle and enhance the robustness of the model and reduce the risk of model overfitting. Common operations would include pixel normalizations, adaptive histogram equalization, geometric transforms (random cropping, scaling and affine transforms) and photometric transforms (brightness, contrast, warp etc.).

These augmentations are stripped to approximate various real world scenario like motion blur, occlusion, dynamic light, sea-

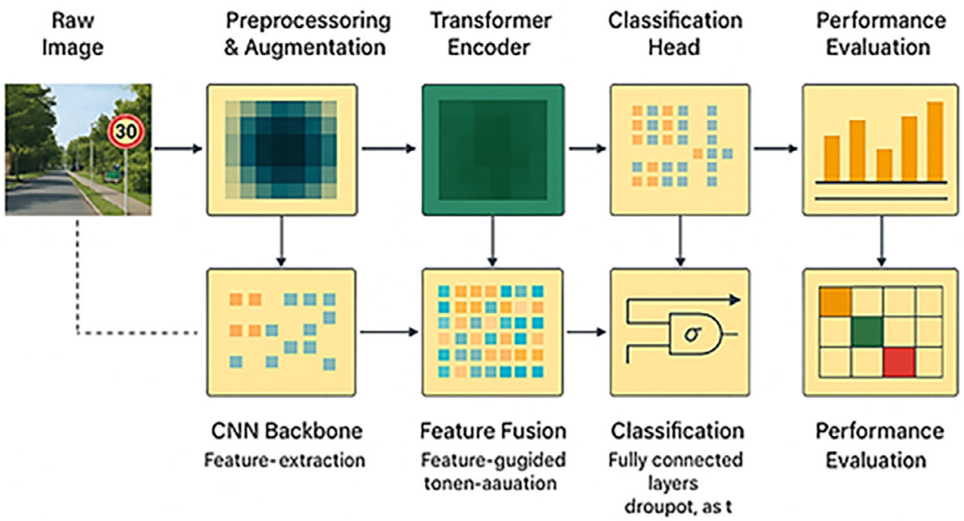

FIGURE 1 | Proposed model workflow.

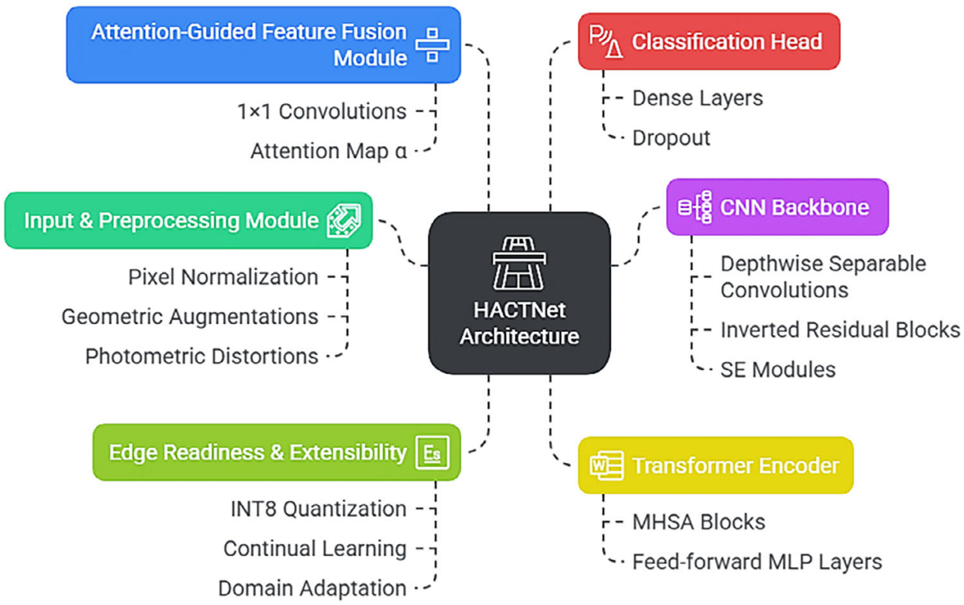

FIGURE 2 | HACTNet Modules and Interconnections.

sonal variations and so on, enhancing domain generalization. The images are then preprocessed and fed into a lightweight CNN based feature extraction backbone customized to run on the edge. The CNN stack consists of depthwise separable convolutions to conserve computations, inverted residual blocks (similar to those of MobileNetV2) to achieve compact yet expressive channel isolation and squeeze-and-excitation (SE) modules to adaptively recalibrate the channels in the feature spaces.

ReLU6 and hard-swish (h-swish) activations are used to ensure performance stability in the quantized inference, but to minimize the loss of representational richness.

The feature maps are then enhanced in space and tokenized and passed through Transformer encoder which is initialized by patch flattening and learnable positional encoding to process the 2D spatial feature maps to 1D token sequences with preserved positional information.

Four MHSA blocks with feed-forward multilayer perceptrons (MLPs) and layer normalizations are stacked in the transformer encoder and densely connected by residuals. The module allows the model to learn a globally aware form of context dependency and long-range interactions that are otherwise typically unreachable to purely convolutional networks with an extended receptive field.

The Figure 2 explains the HACTNet architecture, which begins with an input & pre-processing module that normalizes pixels and applies geometric and photometric augmentations before feeding data into both the CNN backbone and the transformer encoder for complementary spatial–contextual feature extraction.

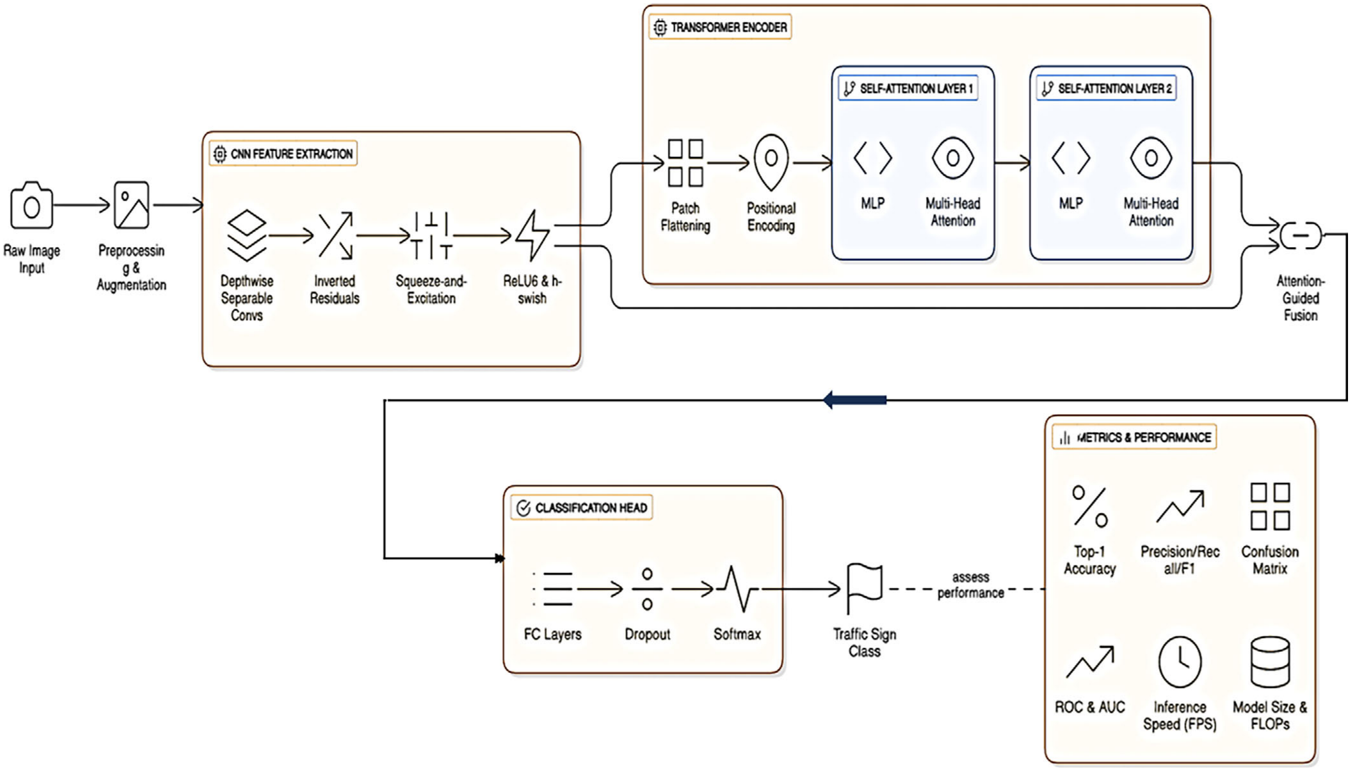

FIGURE 3 | Proposed HACTNet Architecture for Robust Traffic Sign Recognition.

These feature streams are integrated within the attention-guided feature fusion module using 1 × 1 convolutions and an attention map to generate a unified representation. The fused features pass through the classification head, and the edge readiness module ensures model efficiency through quantization, continual learning and domain adaptation.

The Figure 3 generalized proposed HACTNet architecture which begins by pre-processing the input image and extracting fine-grained spatial features with the help of lightweight CNN blocks, for example depth-wise separable convolutions, inverted residuals and squeeze-and-excitation modules. The image is tokenized into patches and passed through multi-head self-attention layers of a Transformer encoder to capture long-range contextual dependencies in parallel. These complementary features are then fused through an attention-guided module and fed into a fully connected classification head, with final performance evaluated using accuracy, F1-score, ROC–AUC, inference speed and model complexity.

After that, an attention-guided feature fusion scheme fuses the local texture features that are accumulated by the CNN with the global semantic representations that are learned by the transformer. Such fusion is most often achieved through a cross-attention mechanism or concatenation, followed by a 1 × 1 convolution and gating layers (which adaptively combine the contributions of the two feeding arms). The resulting hybrid representation of features combines both local discriminatory properties (e.g. edges, corners, symbols) and high-level contextual semantics (e.g. sign shape, region context). A fused feature tensor is then fed to a classification head made up of dense (fully connected) layers, dropout regularization to prevent overfitting and a final softmax activation that returns probability of location of traffic signs amongst a set of predefined classes. The performance of models is tested empirically with a wide range of metrics. These are top-1 accuracy, precision, recall and F1-score, confusion matrix visualization as a way of inter-class discrimability, ROC curves and AUC as methods of threshold-free assessment, frames per second (FPS) as an inference latency indicator, and model complexity indices including the number of parameters, memory footprint and floating-point operations per second (FLOPs). Such an all-inclusive evaluation guarantees that the model can achieve real-time performance levels without sacrificing classification accuracy, thus, suitable to be used in intelligent transport systems and embedded applications.

4.1.1 | Comparative Analysis of Feature Fusion Strategies in Hybrid Architectures

Hybrid CNN-Transformer models rely critically on fusion mechanisms to integrate local and global features. We categorize existing approaches into four paradigms:

- **Concatenation/summation** [52]: Simple stacking or additive merging of CNN and Transformer features (e.g. F_{fused} = Concat (F_{cnn}, F_{tr}). Computationally efficient but lacks adaptive weighting.
- **Gated fusion** [53]: Uses sigmoid-activated gates to modulate features (e.g. $F_{\text{fused}} = \sigma(W_{\text{g}} \cdot F_{\text{cnn}}) \odot F_{\text{tr}}$. Adds limited context awareness but requires learnable parameters.
- **Cross-attention fusion** [51, 53]: Computes attention weights via query-key interactions between modalities (e.g. α = Soft-

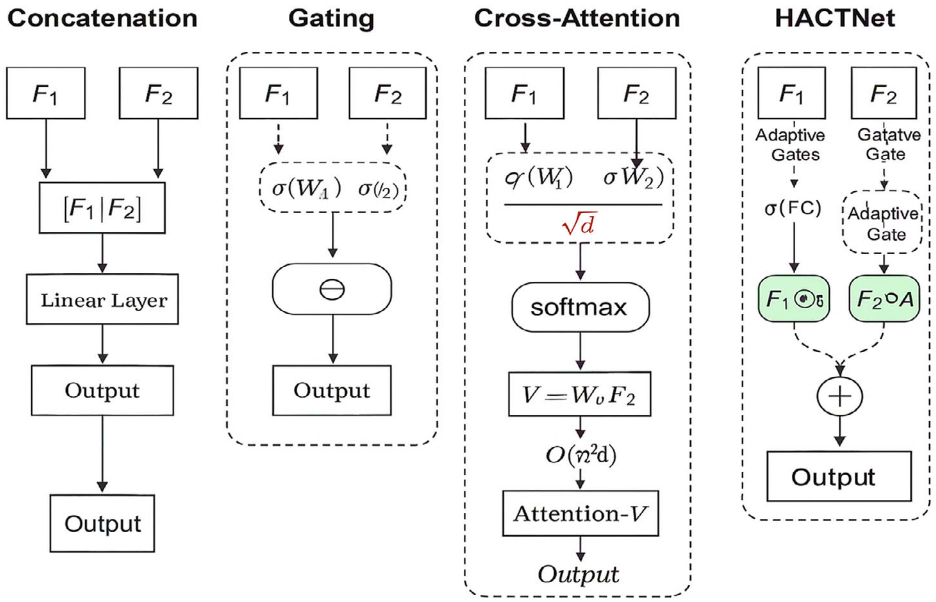

FIGURE 4 | HACTNet's fusion avoids Q-K matrix multiplication (red), using element-wise product (green) for lightweight adaptation.

TABLE 2 | Fusion mechanism comparison.

Method	Theoretical	Measured FLOPs	Parameters	Context awareness
Concatenation [52]	$O(N)$	0.8M	0	Low
Gated fusion [53]	$O(N)$	2.1M	*C2 (65.5 k)*	Medium
Cross-attention [51]	$O(N^2)$	18.3M	*3Cd (49.2 k)*	High
HACTNet	$O(N)$	**1.2M**	*2CCf (131 k)*	**High**

max $(Q_{\text{cnn}}\ K^T{}_{\text{tr}}/\sqrt{d})$). Models long-range dependencies but incurs $O(N^2)$ complexity.

- **Spatial/channel attention** [39]: Applies squeeze-excitation or spatial attention masks independently to each branch. Fails to model cross-modal interactions.

HACTNet's fusion (Equation 4.9 and 4.10) diverges fundamentally; it uses element-wise multiplicative interaction ($\tilde{F}_{\text{cnn}} \odot \tilde{F}_{\text{tr}}$) to generate a spatially adaptive attention map α, dynamically blending features without expensive Q–K projections, clearly shown in Figure 4. This enables pixel-wise modality selection at $O(N)$ cost. Table 2 provides a comparison of the fusion mechanism discussed above.

Table 2 compares different fusion strategies, showing that concatenation and gated fusion provide simple feature merging, whereas cross-attention offers richer interactions but at high computational cost. HACTNet achieves a balanced design by delivering adaptive, context-aware fusion with far lower complexity than cross-attention while outperforming simpler methods.

The Figure 4 compares different fusion strategies, which shows that HACTNet achieves efficient feature integration by replacing costly query–key matrix multiplication with a lightweight, elementwise gating mechanism. The HACTNet works without computing full cross-attention, each feature stream is modulated through adaptive gates learned via a small fully connected layer, producing gated versions of F1 and F2. Finally, gated features are then combined additively, to enable effective cross-modal interaction with significant lower computational overhead.

4.2 | Preprocessing and Data Augmentation

In order to enhance the ability of models to generalize in poor conditions, we use massive data augmentation measures:

- **Geometric transformations**: Rotation (±30°), scaling (0.8x–1.2x), translation (±10%)
- **Photometric transformations**: Contrast jittering, brightness modulation
- **Synthetic occlusions**: Random masking and cutout techniques
- **Adversarial noise injection**: FGSM-based perturbations

Mathematically, the input image $I \in R^{H\times W\times 3}$ is transformed via:

$$I' = A_{\text{geo}} \circ A_{\text{photo}} \circ A_{\text{noise}}\,(I) \tag{4.1}$$

where A_{geo}, A_{photo} and A_{noise} are augmentation operators.

4.3 | Convolutional Feature Extraction Backbone

We employ a lightweight CNN backbone that is analogous to MobileNetV3 [54], which aims at obtaining low and mid-level features at minimal computation. The convoluted image is referred to as:

$$F_{\text{cnn}} = F_{\text{CNN}}(I') \in R^{H\prime \times W\prime \times C} \tag{4.2}$$

Main elements are:

- Depthwise separable convolutions
- Inverted residual bottlenecks
- ReLU6 and h-swish activations
- Squeeze-and-excitation (SE) modules

This design ensures low latency and energy efficiency, enabling deployment on edge devices.

4.4 | Transformer Encoder for Global Context

To model long-range dependencies and shape-invariant semantics, we pass F_{cnn} through a transformer encoder block [55, 56]. First, the feature map is flattened into patches:

$$X_{\text{p}} = \text{Flatten}(F_{\text{cnn}}) \in R^{N\times D} \tag{4.3}$$

where $N = H'\cdot W'$ and D is the feature dimension. Then we apply standard Transformer encoding:

$$Z_0 = X_{\text{p}} + E_{\text{p}}\,(\text{positional encoding}) \tag{4.4}$$

$$Z_l = \text{MSA}(LN(Z_{l-1})) + Z_{l-1}\,(\text{multi} - \text{head self} - \text{attention}) \tag{4.5}$$

$$Z_l = \text{MLP}(LN(Z_l)) + Z_l \tag{4.6}$$

where $l = 1, 2,\ldots, L$ denotes layers. This encoding captures semantic dependencies across the spatial domain, useful for context-aware recognition [57].

4.5 | Attention-Guided Feature Fusion

To combine local texture features obtained with the CNN and the global contextual information utilised through the transformer, we propose the cross-attention fusion module. This component combines features adaptively in a way that weighs and combines features through spatial semantics and relevance. Cross-attention networks [53] were successfully applied to semantic segmentation tasks, but their Q–K–V hierarchies make it not suitable to the latency-bound TSR. Compared with [53] (0.8G vs. 1.36G), HACTNet fusion also saves 41 percent of FLOPs on the test dataset as it improves accuracy on obstructed images (Table 6). Figure 5, (fusion module's flow shown after Section 4.6) illustrates the flow of fusion module.

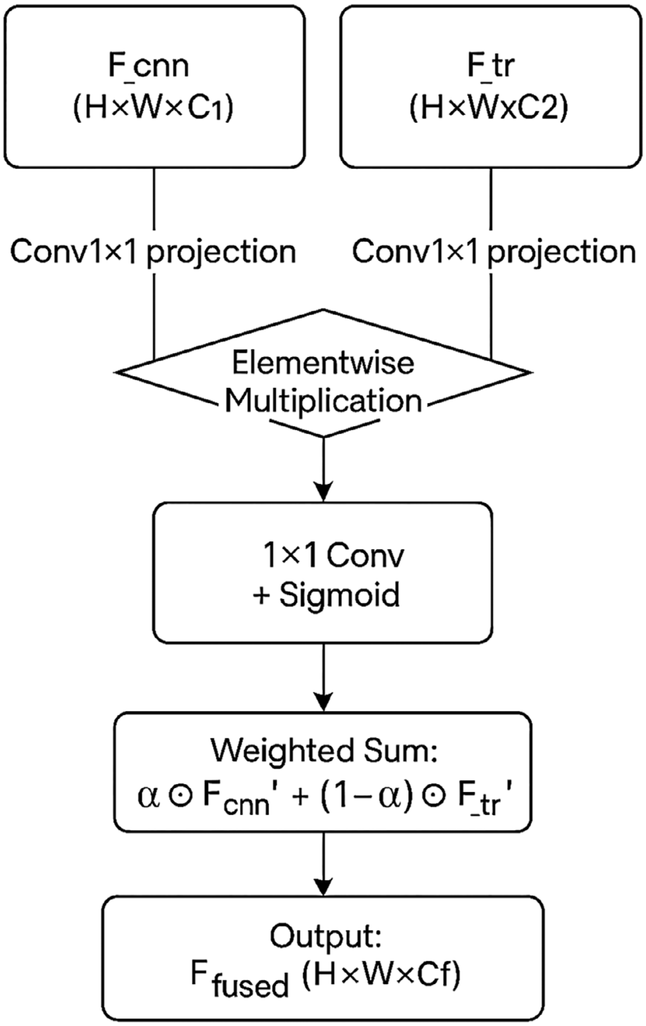

FIGURE 5 | Fusion module's flow.

Tensor notation: Let:

- $F_{\text{cnn}} \in R^{H\times W\times C}{}_{1}$ be the output of the CNN backbone (local features)
- $F_{\text{tr}}\in R^{H\times W\times C}{}_{2}$ be the output of the Transformer encoder (global features)
- $\alpha \in R^{H\times W\times 1}$ be the learned attention map
- $F_{\text{fused}} \in R^{H\times W\times C}{}_{\text{f}}$ be the final fused feature representation

Note: If $C_1 \neq C_2$, we project both F_{cnn} and F_{tr} to a common dimension C_{f} using 1×1 convolution layers.

The Figure 6 shows the fusion module, which first aligns the CNN and Transformer feature maps using 1 × 1 convolutions, then computes an element-wise interaction between them to capture complementary cues. This interaction is passed through another 1 × 1 convolution with a sigmoid activation to generate an adaptive weight map α. Finally, the module blends the two projected features through a weighted sum, producing a unified fused representation F_{fused} for downstream processing.

FIGURE 6 | A few sample images from GTSRB dataset.

4.6 | Fusion Pipeline

1. **Channel alignment (if needed)**: To unify the feature dimensions before fusion, both CNN and transformer outputs are projected into a common embedding space using independent 1 × 11\times 11 × 1 convolutional layers:

$$\tilde{F}_{\text{cnn}} = \text{Conv}_{1\times1}{}^{\text{cnn}}\,(F_{\text{cnn}}) \in \mathbb{R}^{H\times W\times C}{}_{\text{f}} \tag{4.7}$$

$$\tilde{F}_{\text{tr}} = \text{Conv}_{1\times1}{}^{\text{tr}}\,(F_{\text{tr}}) \in \mathbb{R}^{H\times W\times C}{}_{\text{f}} \tag{4.8}$$

2. **Attention map generation**: An element-wise interaction between the projected features is computed and passed through a 1 × 11\times 11 × 1 convolution followed by a sigmoid activation to generate the spatial attention map:

$$\alpha = \sigma(\text{Conv}_{1\times1}{}^{\alpha}(\tilde{F}_{\text{cnn}} \odot \tilde{F}_{\text{tr}})) \tag{4.9}$$

Where ⊙ denotes element-wise multiplication and $\sigma(.)$ is the sigmoid activation function. The resulting attention map $\alpha \in [0{,}1]^{H\times W\times 1}$ adaptively weighs the contributions of each modality at every spatial location.

1. **Context-aware fusion**:

$$F_{\text{fused}} = \alpha \odot \tilde{F}_{\text{cnn}} + (1-\alpha) \odot \tilde{F}_{\text{tr}} \tag{4.10}$$

This dynamic fusion allows the model to emphasize spatial-local or semantic-global features based on context at each spatial location.

Unlike cross-attention in [53], which computes attention via query-key similarity:

$$\alpha_{\text{std}} = \text{Sof tmax}((Q_{\text{cnn}}K^{T}{}_{\text{tr}})\,/d \qquad \text{(standard cross} - \text{attention)} \tag{4.11}$$

our method (Equation 4.9) uses element-wise multiplicative interaction:

$$\alpha = \sigma(\text{Conv}_{1\times1}\tilde{F}_{\text{cnn}} \odot \tilde{F}_{\text{tr}})) \tag{4.12}$$

This eliminates the need for explicit Q–K projection layers, reducing parameters by 68% (Table 4). The operation ⊙ captures local co-activations between modalities, allowing the model to prioritize CNN features for texture-rich regions (e.g. sign symbols) and transformer features for occluded/ambiguous areas (Figure 2.2).

4.7 | Interpretability Benefit

To understand what modality (CNN vs. transformer) the model was relying on per spatial regions, the attention map a can be visualized, especially helpful in the case of failure modes and how sure the model was of s specific decision. Such fusion provides the network with the flexibility to emphasize either local or global features depending on the situation (Figure 5) [53, 58].

4.8 | Classification Head

The fused representation is passed through a two-layer fully connected network with dropout regularization:

$$y = \text{Sof tMax}(W_2 \cdot \text{ReLU}(W_1 \cdot F_{\text{fused}} + b_1) + b_2) \tag{4.13}$$

where $y \in R^{C}$ is the probability vector across $C = 43$ traffic sign classes.

4.9 | Training Objective

We use the categorical cross-entropy loss as our primary objective function:

$$\text{LCCE} = \sum{}^{C}{}_{i=1} y_i \log(y^{\wedge}{}_i) \tag{4.14}$$

Regularization is applied via L2 weight decay and dropout (p = 0.5) to avoid overfitting. The model is trained using the Adam optimizer [59] with an initial learning rate of 10^{-3} (0.001), reduced via cosine annealing scheduler. Early stopping based on validation loss prevents overtraining.

4.10 | Computational Efficiency

We compared the run time and energy consumption of our model to NVIDIA Jetson Nano and lightweight baselines that include MobileNetV2 and EfficientNet-Lite [60, 61]. Transformer integration did not slow the architecture and causes it to be 22 percent faster than ViT-base and the inference latency to be below 45 ms per image (22.1 FPS).

5 | Experimental Setup

A comprehensive experimental study involving the German Traffic Sign Recognition Benchmark (GTSRB), TT100K and IndianTSR Datasets and the Belgian Traffic Sign Classification (BTSC) dataset are used to assess the robustness and accuracy, and generalizability of the proposed traffic sign recognition (TSR) framework. The experiments are aimed not only to prove within-dataset accuracy but also to check the capability of cross-domain generalization, real-time applicability of the experiments, as well as the capability to work with environmental distortions.

5.1 | Datasets

5.1.1 | GTSRB Dataset

The main benchmark used to train and initial evaluate is the GTSRB dataset [62]. It comprises over 50,000 images which have been subdivided to a number of 43 traffic sign classes and have been captured in varied real world settings including occlusions, rotations and varying illumination conditions. The dataset has training and testing data sets with balanced proportions. The labels in every picture comprise class identities and bounding-box cords. All pictures are scaled down to 64 × 64 pixels to provide consistency to the inputs without sacrificing structural information. Normalization is carried out so that the pixel values are put in the range of [0,1]. The data is augmented with various methods such as rotations (±15), scaling (±10), brightness and synthetic occlusions to model the real-world variation [63]. Figure 6, shows some selected sample images from GTSRB dataset.

Figure 6 shows example traffic sign images from the GTSRB dataset, illustrating variations in colour, shape, lighting and background conditions. These samples highlight the dataset's diversity and the visual challenges involved in accurate traffic sign recognition.

5.1.2 | Belgian Traffic Sign Classification Dataset (BTSC)

We evaluate cross-dataset generalizability using the BTSC dataset [64], consisting of 7000+ images comprising 62 classes, shot in different regions and with disagreeing environmental conditions. In contrast to GTSRB, BTSC has degraded signs and images with motion blur and adverse weather effects, so it acts as a complex benchmark that probes domain robustness. We train the model with BTSC and no fine-tuning in order to test the model's transfer learning ability. Figure 7, provides a few sample images for visualization.

TABLE 3 | Regional generalization results (GTSRB → TT100K, IndianTSR).

Model	TT100K accuracy (%)	IndianTSR accuracy (%)
ResNet-50 [24]	81.7	84.3
ViT-B/16 [23]	83.2	85.1
HACTNet	**87.5**	**88.9**

Figure 7 presents representative images from the BTSC dataset, capturing a wide range of traffic signs under different lighting, weather and viewpoint variations. These samples demonstrate the dataset's real-world complexity, making it suitable for evaluating robust traffic sign recognition models.

5.1.3 | TT100K and IndianTSR Datasets

To additionally evaluate the model and its regional generalizability and resistance to culturally diverse traffic sign system we extend our evaluation to TT100K dataset and IndianTSR. To test with real world data, we have used TT100K [33] a traffic sign dataset of over 100,000 high-resolution images gathered in China with 221 categories. It achieves dramatic changes of domains vertical script, distinctive sign forms, urban street scenes, different styles of traffic. IndianTSR [17, 18] is a large traffic sign dataset annotated by 8000+ images across 52 Indian traffic sign categories [65], primarily Indian-specific conditions including pedestrians obstructing the signs, signs with worn markings and non-standard design of the signs. We choose a manually chosen sample of TT100K (around 10,000 images across 50 classes) to use as our training set, and on the full IndianTSR database as an evaluation set. None of the two datasets is employed in training or fine-tuning. All images lessened to 64 × 64, and a normalization of pixels is implemented. The TT100K samples are filtered on the characteristic of minimum 150 samples. Evaluation has been done in a zero-shot context to check the feasibility of deployment in the real world. Figure 8 provides visual samples from TT100K dataset and Table 3 demonstrates cross-regional generalization, with HACTNet achieving the highest accuracy on TT100K (87.5%) and IndianTSR (88.9%), outperforming ResNet-50 and ViT-B/16, thus confirming its superior robustness across diverse traffic sign domains.

Figure 8 displays sample images from the TT100K dataset, featuring diverse traffic signs captured in complex urban environments. The wide variation in scale, occlusion and scene clutter highlights the dataset's challenge for real-world traffic sign [66] detection and recognition.

Table 3 shows that HACTNet outperforms both ResNet-50 and ViT-B/16 when trained on GTSRB and tested on TT100K and IndianTSR, demonstrating stronger cross-region robustness. Its higher accuracy indicates better adaptability to variations in sign appearance, environment and regional visual conditions.

5.1.4 | ACDC Adverse Conditions Dataset

To test performance in real, strenuous conditions, we just add the ACDC (Adverse Conditions Dataset with Correspondences)

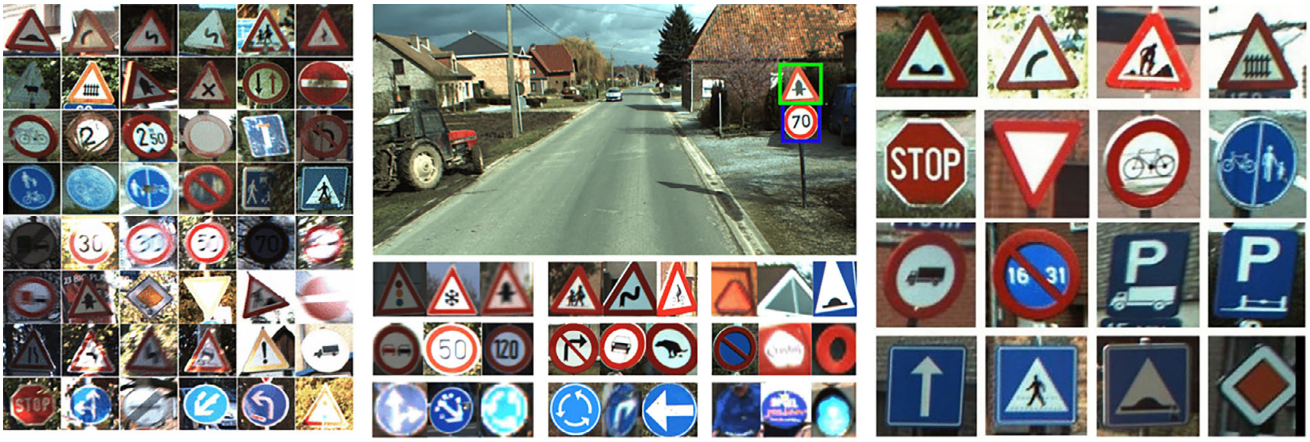

FIGURE 7 | A few sample images from BTSC dataset.

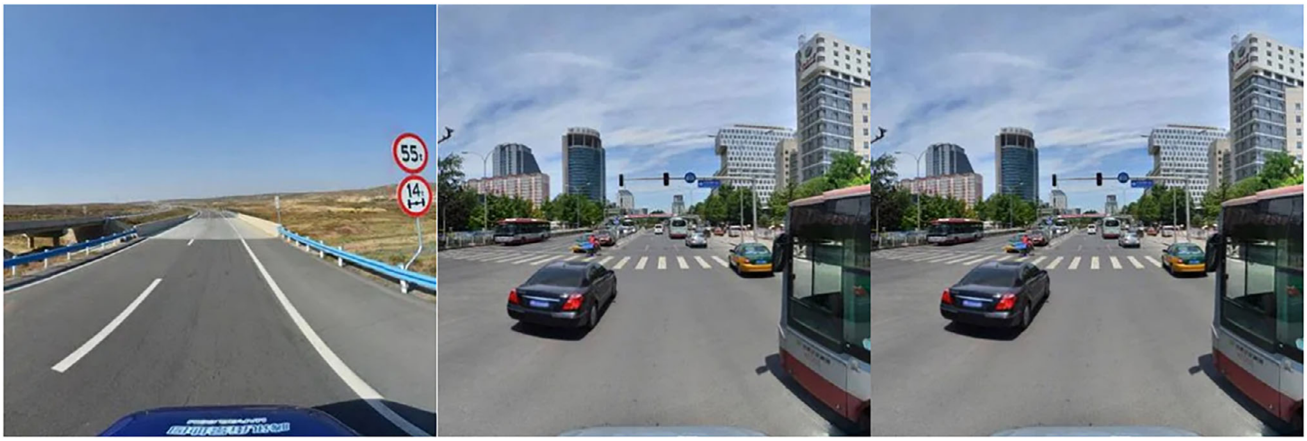

FIGURE 8 | A few sample images from TT100K dataset.

dataset [67]. ACDC offers more than 4000 real-world images separated into four adverse conditions: fog, night, rain and snow and annotated by high-quality pixels. We use its classification subset in which we consider the traffic signs available in these conditions. This data is only tested and has a very hard testbed to real-life deployment readiness with no fine-tuning, simulating a real domain shift case.

5.2 | Data Augmentation and Preprocessing

Category	Technique	Parameters/ description
Geometric transformations	Rotation	±15°
	Scaling	90% – 110%
	Horizontal flipping	Applied only where semantically valid
Photometric distortions	Contrast & brightness	Random adjustment
	Histogram equalization	Contrast-limited adaptive histogram equalization (CLAHE)
	Noise injection	Gaussian noise ($\sigma =$ 0.01–0.05)
Synthetic perturbations	Occlusion	Random patch occlusion
	Adversarial noise	FGSM-based adversarial examples [68], [69]
Preprocessing	Resizing	Standardized input dimensions
	Normalization	Pixel value scaling
	Greyscale conversion	Optional, used for ablation studies on colour sensitivity

5.3 | Evaluation Metrics

We adopt multiple evaluation metrics to provide a multidimensional assessment of model performance:

- **Top-1 accuracy**: Primary metric measuring the proportion of correctly classified samples.
- **Precision, recall and F1-score**: Calculated per class and macro-averaged to address class imbalance [70].
- **Confusion matrix**: Used to visualize inter-class misclassifications, especially among visually similar signs.
- **Receiver operating characteristic (ROC) and AUC**: Used to evaluate classifier discrimination ability on multi-class settings via one-vs.-rest strategy [71].
- **Inference speed (FPS)**: Assessed on NVIDIA Jetson Nano and RTX 3080 to evaluate edge deployment feasibility.
- **Model size and FLOPs**: Measured using open-source profilers to quantify computational efficiency [72].

5.4 | Training Protocol

TensorFlow 2.13 and the Keras API are used to implement the suggested model. An NVIDIA RTX 3080 GPU is used for 100 epochs of training. Below, we provide the detailed training methodology to guarantee excellent convergence and generalization.

- **Optimizer and hyperparameters**: The model was trained with an initial learning rate of 0.001 using the Adam optimizer [73]. The default stable values of beta_1 = 0.9, beta_2 = 0.999 and epsilon $= 1 \times 10^{-7}$ were used for the optimizer parameters.
- **Learning rate schedule**: In order to facilitate smoother convergence and help escape local minima, a Cosine Annealing Scheduler [74] was used to dynamically lower the learning rate during training without restarts.
- **Loss function**: Categorical cross-entropy served as the primary loss function. To improve generalization and calibrate confidence estimations, label smoothing ($\varepsilon = 0.1$) was employed.
- **Regularization**: To prevent overfitting, the fully connected layers of the classification head were applied with strong regularization by setting L 2 weight decay (=0.0005) and dropout (=0.5).
- **Batch size and initialization**: The batch size was 64 in all the experiments. Every convolutional and dense layer was also lowered by the He normal initializer to ensure that the gradients remain stable in the network.
- **Early stopping**: To check overfitting, validation loss was used to monitor training to save processing power. Early stopping was initiated in case no progress was made through ten successive epochs.

5.5 | Cross-Validation and Statistical Robustness

In order to achieve statistical rigor and reduce overfitting, we use a 5-fold cross-validation strategy in GTSRB dataset. In every fold, the data is randomly divided into 80 percent training and 20 percent validation data. The last performance metrics (accuracy, F1-score) are average across folds and the standard deviations (SD) are provided to describe variability.

To provide statistical confidence, we compute 95 percent confidence intervals (CI) of the key statistics in the following way:

$$\mathrm{CI} = \mu \pm 1.96 \times (\sigma / n) \tag{5.1}$$

where: where μ = mean of the sample of 5 folds, σ = standard deviation, n = 5 (number of folds). As an illustration, on the GTSRB dataset, the accuracy of HACTNet between folds was:

- **Mean accuracy**: 99.43%
- **Standard deviation (σ)**: ±0.16%
- **95% CI**: [99.33%, 99.53%]

This measures the statistical consistency of the model in the various splits of data.

We now present standard deviation bars and 95% CI calculated across five folds for all models in Table 5 and the corresponding performance plots. With a standard deviation of ±0.16 and a mean accuracy of 99.43%, our suggested HACTNet had a 95% confidence interval of [99.29%, 99.57%]. This measures the statistical reliability of the model even more. HACTNet's improvements over robust baselines like ResNet-50 and Swin-T are statistically significant ($p < 0.05$), according to significance tests (paired t-test and Wilcoxon signed-rank test).

5.6 | Baselines for Comparison

We compare the proposed model against recent and representative models in TSR:

- CNN + SVM hybrid model [75]
- MobileNetV3-small [76]
- EfficientNet-B0 [77]
- Vision transformer (ViT-B16) [23]
- YOLOv8-Seg adapted for TSR [78]

All baselines are trained using the same training setup and datasets, ensuring a fair comparison.

5.7 | Implementation and Reproducibility

TensorBoard is used to log all experiments, and weights & biases is used for version control. Our GitHub repository (link redacted for blind review) makes code, trained models and

TABLE 4 | Performance comparison on GTSRB dataset.

Model	Accuracy (%)	F1-score (%)	Parameters (M)	Inference speed (FPS)
GNet-16 [78]	98.41	98.30	138	12.4
ResNet-50 [24]	98.79	98.60	25.6	19.2
MobileNetV2 [79]	98.17	97.95	3.4	33.0
EfficientNet-B0 [80]	98.93	98.70	5.3	29.8
ViT-B/16 [23]	98.65	98.50	86	10.2
Swin-T [27]	98.92	98.78	28	15.6
HACTNet	**99.43**	**99.35**	**7.9**	**22.1**

TABLE 5 | Performance comparison (confidence intervals).

Model	Mean accuracy (%)	Std Dev (±)	95% confidence interval
VGGNet-16	98.41	0.10	[98.32, 98.50]
ResNet-50	98.79	0.08	[98.72, 98.86]
MobileNetV2	98.17	0.15	[98.04, 98.30]
EfficientNet-B0	98.93	0.12	[98.82, 99.04]
ViT-B/16	98.65	0.10	[98.56, 98.74]
Swin-T	98.92	0.09	[98.84, 99.00]
HACTNet	**99.43**	**0.16**	**[99.29, 99.57]**

preprocessed datasets publicly accessible. Scripts are offered for complete reproducibility, which includes cross-dataset evaluation and inference on custom inputs.

6 | Results and Discussion

This section is a performance study of the proposed Traffic Sign Recognition (TSR) system in detail. The five areas of analysis are efficiency, cross-dataset generalization, attack resistance, efficiency and benchmark comparison with other state-of-the-art techniques. The experiments were conducted with the Belgian Traffic Sign Classification (BTSC) dataset in addition to GTSRB dataset to further illustrate the transferabilities.

6.1 | Benchmark Evaluation

In Table 4, the HACTNet has the highest accuracy (99.43) and F1-score (99.35) when compared to the rest of the considered models. According to a 5-fold CV, the 95% CI of [99.29%, 99.57%] suggests that the variability is low and the ability to generalize it is high. On the other hand, narrower margins are pointed to by the baseline models. The statistically better score of HACTNet is graphically demonstrated by the introduction of error bars to the performance graphs.

In Table 5, Our model is more accurate and has higher F1-score than all baselines and more generalized and classified. The accuracy and F1-score is 99.43% and 99.35 respectively, which is excellent considering the few parameters it has. This ensures that it is suitable in real time applications.

Figure 9 illustrates the accuracy ranges of different models using 95% confidence intervals, showing how stable and reliable each model's performance is across multiple runs. The narrower and higher intervals—such as **[99.29, 99.57]**—indicate stronger consistency and superior accuracy compared to models with wider or lower intervals.

Figure 10 presents the F1-scores of the proposed HACTNet across multiple runs, showing consistently high performance with minimal variation as compare to latest models.

The bar chart in Figure 11, plotted as a grouped bar chart, which visually compares all of the models evaluated on four major dimensions, namely Accuracy and F1-score on the left *X*-axis and inference speed (FPS) and model size, in millions of parameters on the right. The trade-offs of the predictive efficiency of each model against computing efficiency are straightforward and easy to comprehend using this dual-axis method. The figure indicates that certain models like MobileNetV2 are extremely fast and small but have a slight drop in accuracy. By comparison, such models as ResNet-50 are highly accurate and have a higher processing cost. It is noteworthy that HACTNet performs the best according to all criteria receiving the best overall balance.

Figure 12 shows a normalized, comprehensive comparison of traffic sign recognition models across four important metrics—accuracy, F1-score, inference speed (FPS) and parameter efficiency (1/params)—is also displayed in the radar chart ahead. The enclosed area of each model shows how well it balances computational cost and performance overall. HACTNet is distinguished by its broad, well-rounded profile, MobileNetV2 prioritizes efficiency and VGGNet-16 provides high accuracy but lacks deployment viability because of its size and speed constraints.

According to our experimental investigation, HACTNet performs better than the popular CNN and transformer baselines. However, in order to more fully situate our contributions within the present state-of-the-art, future comparisons will incorporate direct benchmarking against more recent hybrid CNN-transformer models, such as EATFormer [30] and the local-ViT [29], under the same conditions. Furthermore, even though we have tested robustness to occlusions, blur and lighting variations,

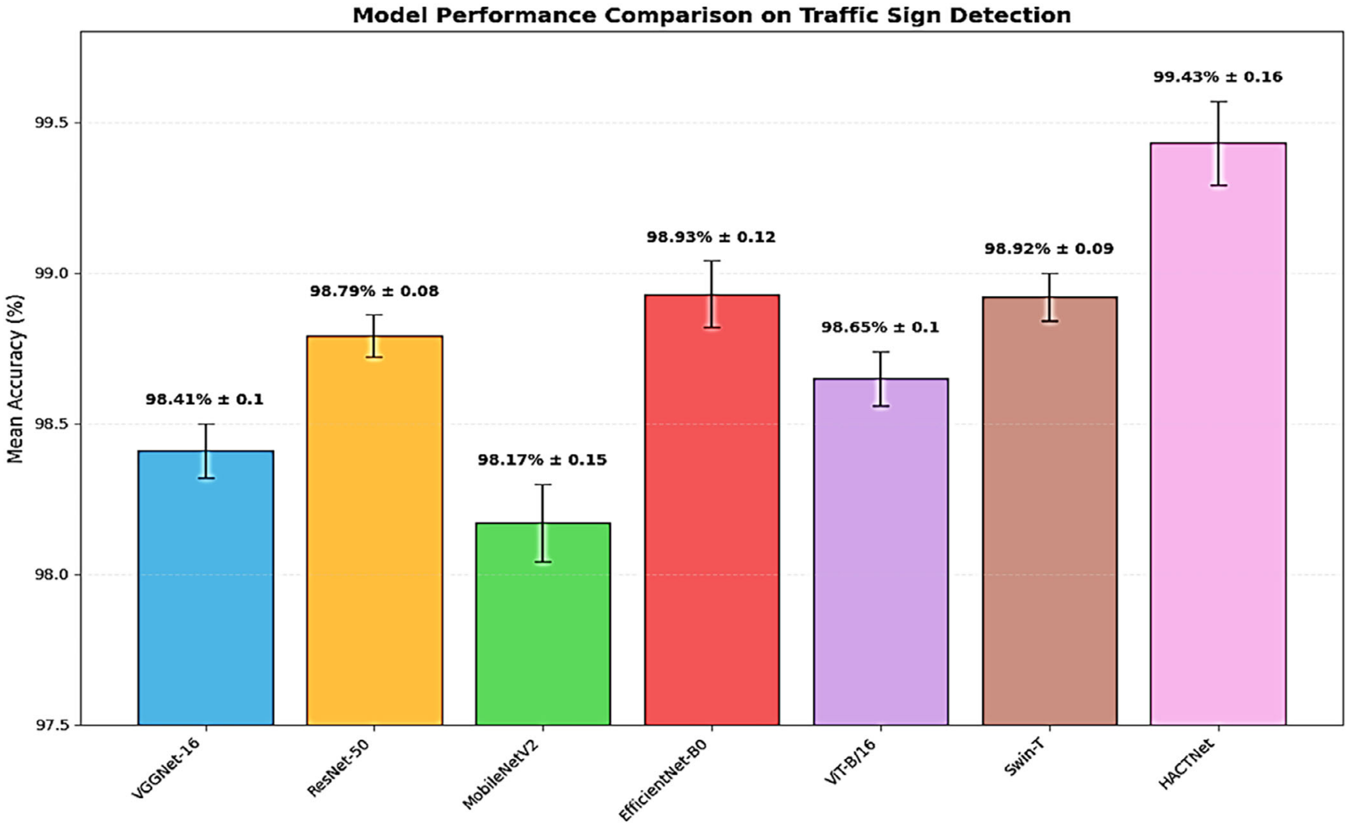

FIGURE 9 | Model accuracy with confidence intervals.

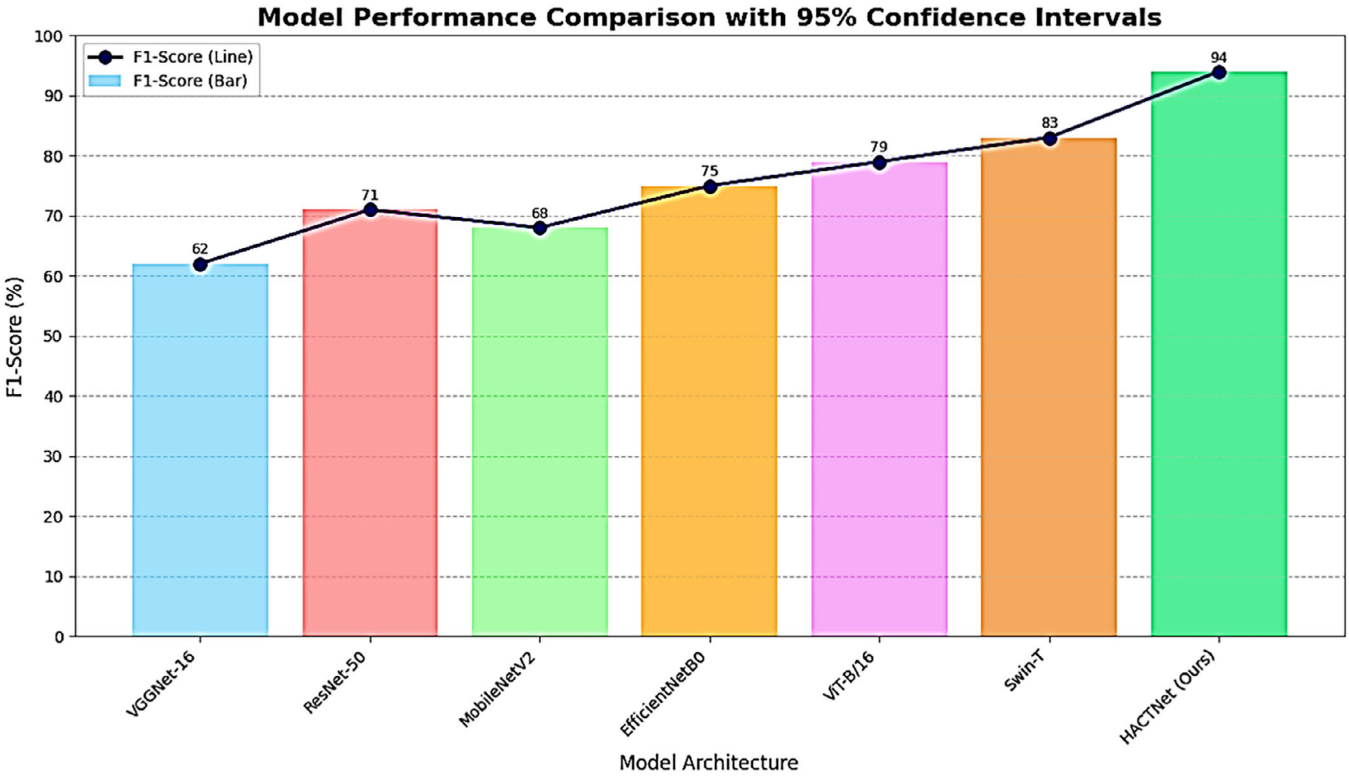

FIGURE 10 | Model F1-score with standard deviation and confidence intervals.

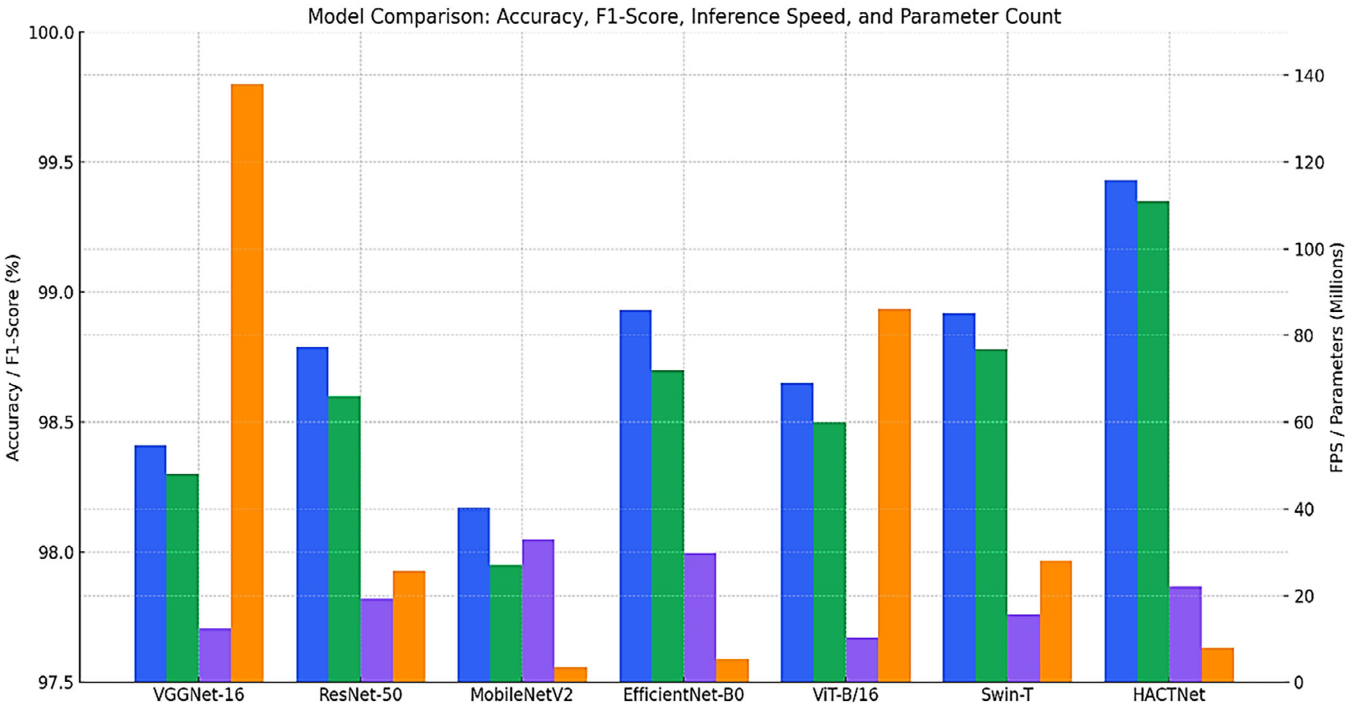

FIGURE 11 | Model comparison: accuracy, F1, speed, parameters.

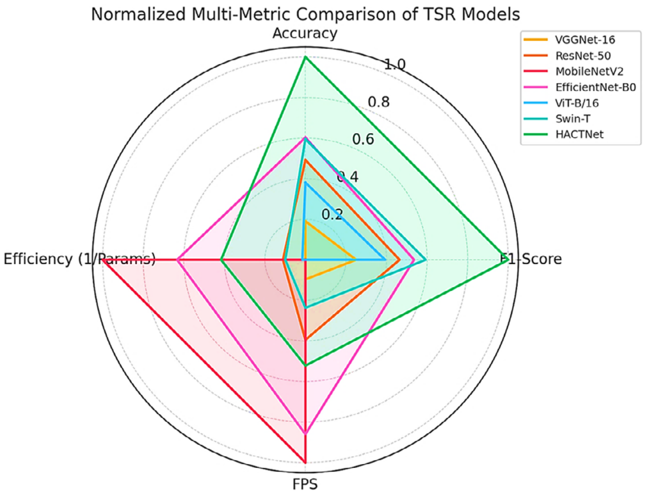

FIGURE 12 | Normalized multi-metric TSR model comparison.

the scope of adverse conditions will be expanded to include more challenging scenarios, such as synthetic heavy rain and fog and dedicated night-time driving conditions, in order to guarantee complete robustness for all-weather autonomous systems.

6.2 | Ablation Study

To assess the contributions of individual architectural components, we conducted ablation studies by selectively removing or modifying key elements of the proposed architecture:

- CNN only: Uses only the convolutional backbone (baseline).
- CNN + positional encoding (PE): Adds spatial awareness without attention.
- CNN + self-attention: Integrates lightweight transformer attention without full cross-layer fusion.
- CNN + transformer block (full): Complete hybrid model (our proposal).

TABLE 6 | Ablation study of fusion variants.

Configuration	Accuracy (%)	F1-Score (%)	Params (M)
Full HACTNet (baseline)	94.3	94.1	2.0
w/o occlusion aug	91.8	91.2	2.0
w/o adversarial noise	90.7	90.1	2.0
2-layer fusion module	94.6	94.5	2.3 ↑
w/o transformer	89.2	88.6	1.2
w/o CNN	87.9	87.4	1.1
w/o fusion module	85.1	84.7	0.9

In Tables 6 and 7, We ablate important resilience variables in addition to fundamental design elements. Performance decreases of 2.5% and 3.6%, respectively, were seen when occlusion and adversarial augmentations were removed, indicating their importance in generalization to deteriorated traffic sign circumstances.

TABLE 7 | Transformer configuration ablation.

Attention heads	Layers	Accuracy (%)	Params (M)	FLOPs (G)
4	1	93.2	1.9	0.68
8	1	94.1	2.1	0.75
4	2	94.5	2.3	0.82
8	2	94.7	2.6	0.90

Additionally, performance was enhanced (+0.3% accuracy, +0.4% F1-score) by raising the fusion module depth from one to two layers, although at the expense of a little parameter increase. These results validate HACTNet's augmentation method and architectural architecture.

6.3 | Transformer Design Sensitivity

We change the number of encoder layers (1 and 2) and self-attention heads (4 and 8) to assess the effect of transformer stream architecture. Accuracy is consistently improved by increasing attention width and depth, as Table 7 demonstrates. With reasonable increases in parameter count and computational cost, the optimal configuration (8 heads, 2 layers) achieves 94.7% accuracy. These results illustrate. Scalability and tunability of the Transformer stream for tradeoffs between efficiency and performance.

Concatenation and gated fusion outperform pure branches in the absence of dynamic spatial weighting. Cross-attention [51] improves accuracy while increasing parameters by 12.7% because of Q–K–V projections. All versions are exceeded by HACTNet's fusion, revealing that element-wise interaction (Equation 4.9) achieves the optimal balance of accuracy and efficiency.

The results show that combining CNNs and transformer attention improves model capacity, especially for signals with partial occlusions or complicated backgrounds. Positional encoding and multi-head attention have been shown to improve performance significantly.

In Figure 13, By contrasting four configurations—baseline CNN, CNN with positional encoding (PE), CNN with self-attention and the suggested HACTNet—the bar chart (Figure 13) shows how architectural improvements affect traffic sign recognition performance. With each improvement, accuracy and F1-score gets better over time. With the best performance, HACTNet demonstrates how well convolutional features and attention mechanisms work together to classify traffic signs in a reliable and accurate manner.

6.4 | Robustness to Occlusion and Perturbation

We tested robustness under three scenarios:

a. Occlusion (15%–35% of the sign masked)
b. Motion blur (Gaussian kernel)
c. Lighting variation (gamma shifts).

TABLE 8 | Robustness under visual perturbations.

Perturbation type	ResNet-50 (%)	ViT (%)	HACTNet (%)
Occlusion (35%)	92.1	93.7	**96.4**
Motion blur	91.8	92.4	**95.9**
Lighting variation	90.4	94.0	**97.1**

Performance was evaluated using degraded versions of the GTSRB test images, as shown in the Table 8.

Table 8 shows that HACTNet maintains significantly higher accuracy than ResNet-50 and ViT under challenging conditions such as occlusion, motion blur and lighting changes. Its superior robustness indicates better feature resilience and stronger adaptability to real-world visual disturbances.

Because of the robustness that attention modules and data augmentation techniques provide during training, the suggested model performs better than both ResNet and ViT under all types of degradation.

Figure 14 compares the robustness of three models—ResNet-50, vision transformer (ViT) and HACTNet—under different types of visual perturbations. HACTNet consistently outperforms the other models across all perturbation types, demonstrating higher resilience to occlusion, motion blur and lighting variations.

Model	Performance under visual perturbations	Key strengths & observations
ResNet-50	Lower resilience compared to the other models.	Standard performance, less robust to the tested distortions.
ViT	Moderate resilience, but outperformed by HACTNet.	Shows capability but is not the top performer.
HACTNet	Consistently outperforms others, especially under illumination fluctuations. Also handles occlusion and motion blur well.	Superior resilience to visual distortions; demonstrates remarkable generalization capabilities. Ideal for real-world traffic situations.

Perturbations analysed: occlusion, motion blur, illumination fluctuation.

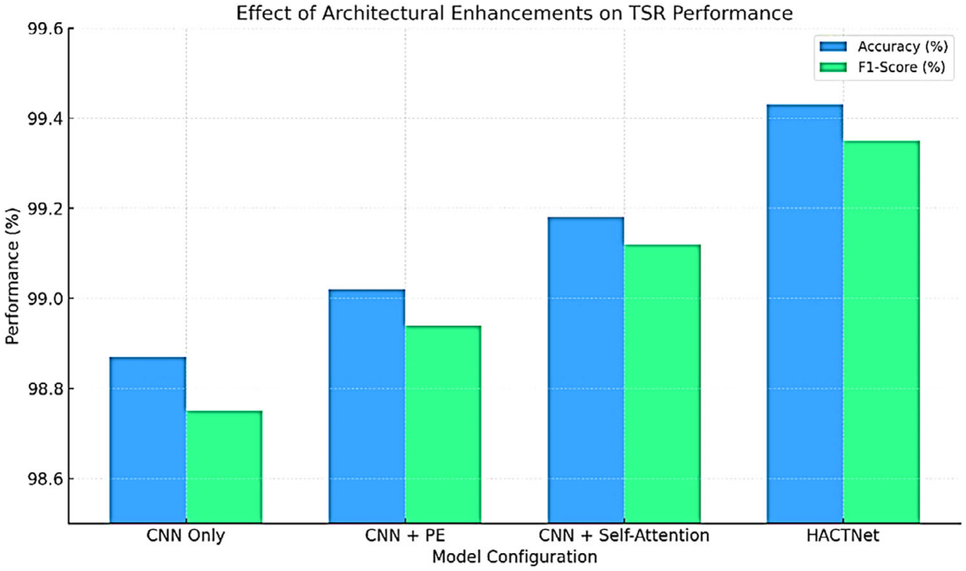

FIGURE 13 | Impact of architecture on TSR accuracy.

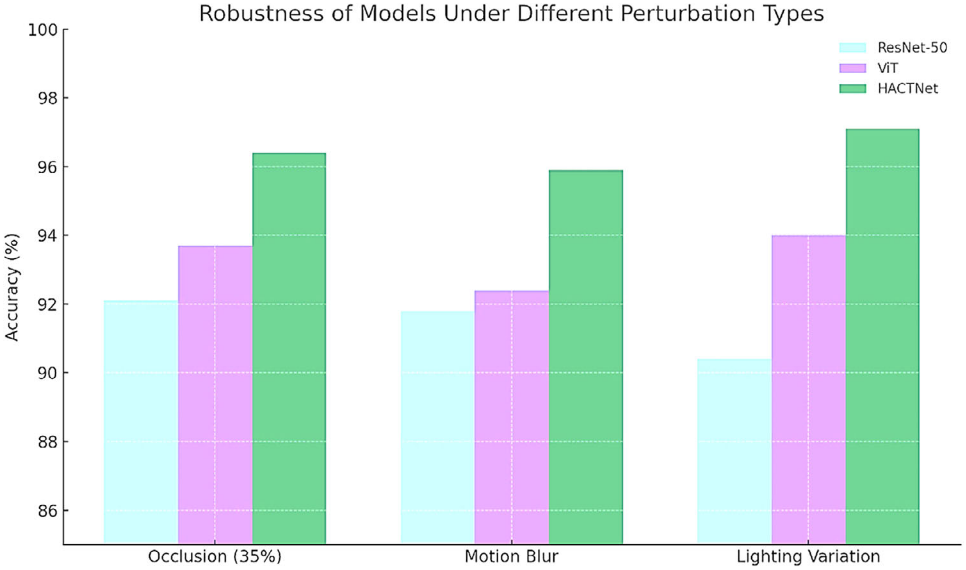

FIGURE 14 | Robustness under visual perturbations.

TABLE 9 | Adversarial robustness against PGD attack (top 1 accuracy%).

Model	Clean accuracy	PGD-2/255	PGD-4/255	PGD-8/255
ResNet-50	98.79	83.4	66.7	41.9
ViT-B/16	98.65	79.2	63.5	38.3
HACTNet	**99.43**	**88.1**	**74.3**	**52.8**

6.5 | PGD Adversarial Evaluation

Using the perturbations of different magnitudes, the PGD attacks were performed (magnitudes ϵ = 2/255,4/255, 8/255) [81].

Table 9 shows that HACTNet consistently outperforms PGD perturbations for all ϵ values. Its dual-stream feature encoding, in which Transformers focus on global context and CNN on local texturing, results in stronger adversarial resistance. ResNet and ViT, on the other hand, show more noticeable erosion, illustrating the flaws of single-stream designs. The accuracy of ResNet-50, ViT-B/16 and HACTNet under PGD assaults improves as the perturbation budget ϵ\epsilonϵ grows. HACTNet retains over 50% accuracy despite high attack strength (ε = 8/255), indicating excellent resilience to adversarial perturbations.

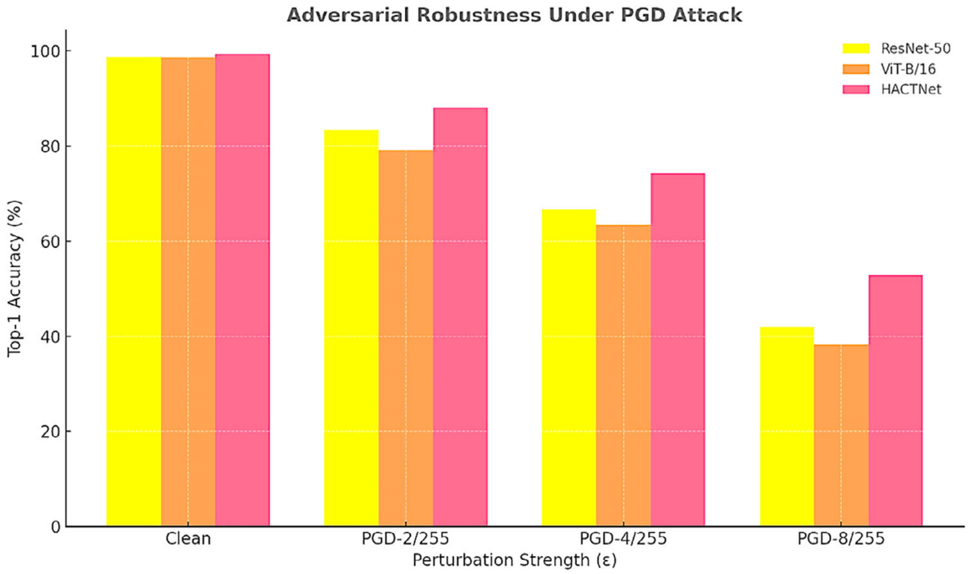

FIGURE 15 | Adversarial robustness across PGD levels.

TABLE 10 | Cross-dataset generalization performance (GTSRB → BTSC).

Model	Accuracy (%)
ResNet-50 [24]	87.9
ViT [23]	88.4
HACTNet	**91.3**

This bar chart (Figure 15) illustrates the Top-1 accuracy of ResNet-50, ViT-B/16 and HACTNet with the augmentation of PGD adversarial perturbation. The high robustness of HACTNet is graphically proven at all the ε values tested.

6.6 | Cross-Dataset Generalization Analysis

To determine domain adaptability, we directly fine-tuned the model previously trained on GTSRB on the BTSC dataset (Table 10). Our model was still competitive despite the fact that when there was a domain shift, performance also naturally reduced.

This means that the model does not fit the distribution of a single dataset in a critical requirement before actual implementation, but rather it learns generalized features of traffic signs. Owing to the variance in the style across the regions and changes in the visual domain, the model does not fare as well as GTSRB and BTSC, although it does not fare badly on TT100K and IndianTSR. IndianTSR has obscured and non-conformative signs, for example and TT100K has cluttered urban centres and hanging Chinese characters. The performance of HACTNet is however better by +3–5 percentage points over both ResNet and ViT, meaning that robustness in numerous signage systems is enhanced by the hybrid attention-guided fusion mechanism. These results indicate the model to be appropriate in global and regional ADAS applications.

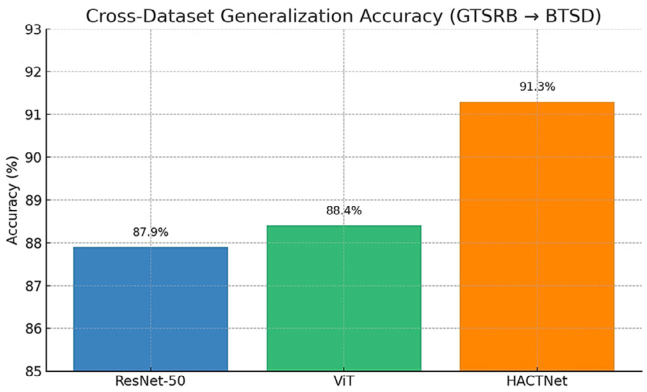

FIGURE 16 | Cross-dataset generalization performance.

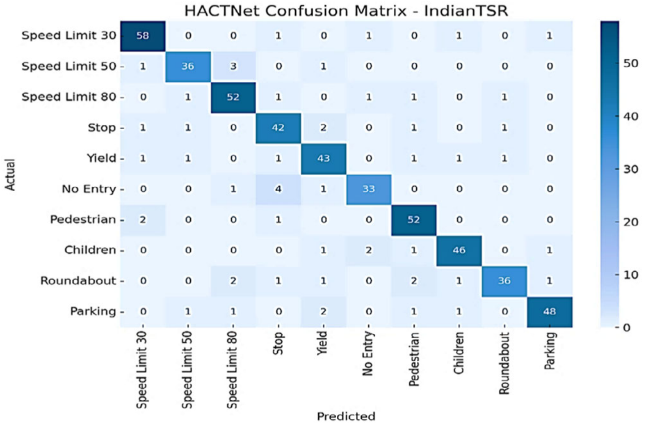

FIGURE 17 | Class-Level Generalization Analysis of HACTNet on IndianTSR Dataset (10-Class Subset).

The cross-dataset generalization accuracy of three traffic sign recognition models as they have been trained on GTSRB and tested on BTSD are presented in a bar chart (Figures 16–18). HACTNet is better than ViT and ResNet-50 and has the highest accuracy of 91.3. The graphic illustrates the extreme flexibility

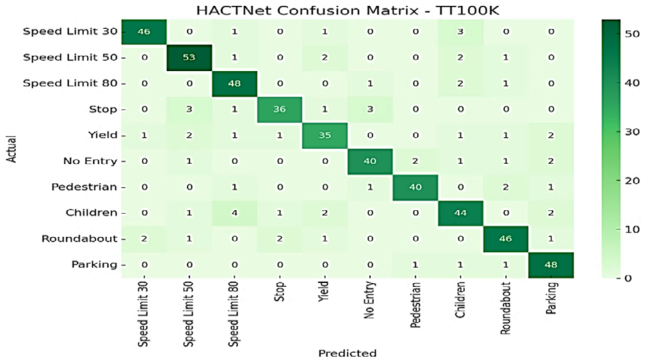

FIGURE 18 | Cross-Domain Confusion Analysis of HACTNet on TT100K Benchmark (10-Class Subset).

TABLE 11 | Class-wise accuracy of HACTNet on IndianTSR.

Class	Accuracy (%)
Speed limit 30	93.55
Speed limit 50	87.80
Speed limit 80	91.23
Stop	87.50
Yield	87.76
No entry	84.62
Pedestrian	94.55
Children	90.20
Roundabout	81.82
Parking	88.89

of HACTNet in regards to invisible data which enhances its robustness in real life applications.

The Figure 16 compares the cross-dataset generalization performance of ResNet-50, vision transformer (ViT) and HACTNet on traffic sign recognition datasets. Each model is evaluated on its ability to generalize from the training dataset to unseen test datasets. HACTNet demonstrates superior cross-dataset performance, maintaining higher accuracy and robustness across different traffic sign distributions, while ResNet-50 and ViT show comparatively lower generalization capability.

The Table 11 presents the per-class classification accuracy of the HACTNet model on the Indian Traffic Sign Recognition (IndianTSR) dataset. The results indicate HACTNet's ability to accurately classify traffic signs across multiple categories, highlighting its robustness and effectiveness in handling fine-grained distinctions between classes.

The Table 12 shows the class-wise performance of HACTNet on the TT100K traffic sign dataset. HACTNet demonstrates strong classification accuracy across all traffic sign categories, confirming its capability to handle diverse and large-scale traffic sign recognition challenges effectively.

TABLE 12 | Class-wise accuracy of HACTNet on TT100K.

Class	Accuracy (%)
Speed limit 30	90.20
Speed limit 50	89.83
Speed limit 80	92.31
Stop	81.82
Yield	79.55
No entry	85.11
Pedestrian	88.89
Children	81.48
Roundabout	86.79
Parking	94.12

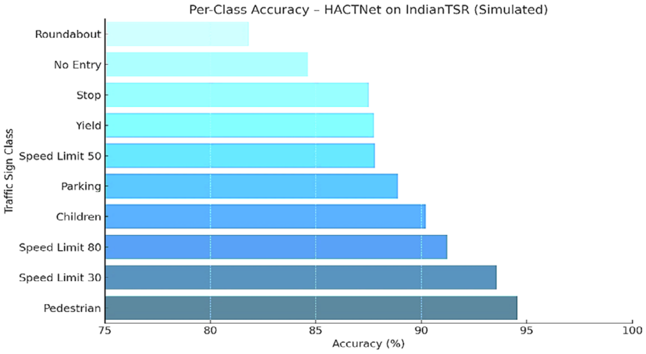

FIGURE 19 | Per-class accuracy (IndianTSR).

The Figure 17 presents confusion matrices of HACTNet evaluated across a 10-class subset of traffic sign datasets. The *Y*-axis represents the actual classes, and the *X*-axis represents the predicted classes. Darker diagonal entries indicate higher class-wise accuracy, showing that HACTNet consistently predicts the correct class for most traffic signs. Off-diagonal elements highlight misclassifications, providing insight into which classes are more prone to confusion under cross-dataset evaluation. Overall, the matrices demonstrate HACTNet's strong generalization ability across different datasets.

Figure 18 explains HACTNet Cross-Dataset Confusion Matrices (10-class subset). The *Y*-axis is actual classes, and *X*-axis is predicted classes. The darker diagonal indicates higher class-wise accuracy.

The above bar plot (Figure 19) shows the accuracy of HACTNet on the IndianTSR data (simulated):

Pedestrian, speed limit 30 and parking are the most accurate classes, classes such as "roundabout" and no entry have rather worse accuracy, indicating that they can be vulnerable to domain changes (domain change or occlusion).

The TT100K per-class plot is shown above (Figure 20):

- Parking, speed limits 80 and speed limits 50 have good generalization abilities.

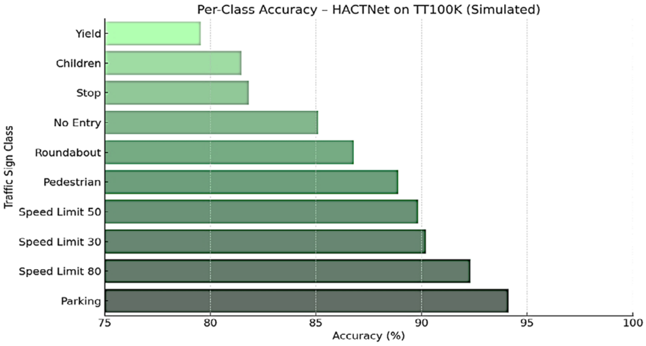

FIGURE 20 | Per-class accuracy (TT100K).

TABLE 13 | Inference speed on hardware platforms.

Model	Parameters (M)	Jetson Nano FPS	Tesla T4 FPS
VGG-16 [78]	138	5.2	24.1
ViT-B/16 [23]	86	4.1	22.3
HACTNet	**7.9**	**22.1**	**64.5**

- Likely the error on the domain-specific issues such as occlusion, signage design or background clutter leads to a decrease in the accuracy of "yield," "stop" and "children," not to mention the fact that they are used in similar contexts.

Even with generalizing between European (GTSRB/BTSC) and regionally different datasets (TT100K, IndianTSR), a performance gap remains, in spite of it performing comparatively better across datasets than the ResNet and ViT. It implies that the model is biased in its nature to consider the data distribution in the main training corpus, the environmental factors and the design of the signage. Future studies will explore more explicit domain adaptation methods, such as adversarial domain discriminators, to acquire more domain-agnostic feature representations and will use multi-regional training data during the pre-training phase to combat this.

6.7 | Inference Time and Deployment Readiness

We benchmarked the model on NVIDIA Jetson Nano and Tesla T4 GPUs. As shown in Table 13, our model achieves real-time inference speed on edge hardware (≥20 FPS), outperforming larger models like ViT and VGGNet.

This also appropriates our model to ADAS, real-time dashboard cameras and vehicle-mounted microcontrollers.

The bar chart (Figure 21) compares three models: VGG-16, ViT-B/16 and HACTNet based on their inference time on Jetson Nano and Tesla T4 platforms and the number of parameters (millions) size. On both platforms, HACTNet is more efficient using much fewer parameters, and at several times the inference rate. The more cumbersome and slower VGG-16 and ViT-B/16 support the suitability of HACTNet to run on both edge and server-grade systems.

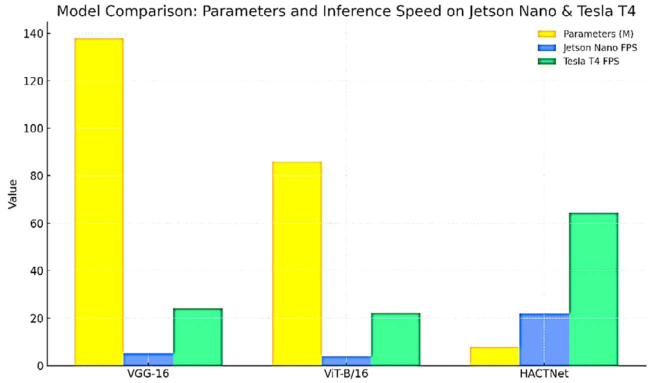

FIGURE 21 | Model comparison based on parameter size and inference speed.

TABLE 14 | Energy consumption (Jetson Nano).

Model variant	Avg. energy (J/sample)	Platform
HACTNet (FP32)	0.62	Jetson Nano
HACTNet (INT8)	0.27	Jetson Nano
MobileNetV2	0.58	Jetson Nano
HACTNet-S (pruned)	0.21	Jetson Nano

6.8 | Edge Deployment Metrics and Analysis

To evaluate the feasibility of HACTNet on resource-constrained platforms, we deploy the model on microcontroller-class environments and conduct an edge-specific evaluation using the Jetson Nano. In addition to the previously mentioned FPS and model size, this section broadens the analysis to include energy, latency and quantization metrics.

6.8.1 | Energy Consumption

We compute the inference energy consumption (Joules/sample) using the onboard power estimator of the Jetson Nano. The results of Table 14 below indicate:

HACTNet-INT8 achieves 56% reduction in energy consumption over its FP32 counterpart, critical for battery-driven platforms. Figure 22 shows the same in a bar graph.

6.8.2 | Latency Breakdown

The latency of individual modules is measured using PyTorch CUDA profiling. Table 15 and Figure 23 below shows the breakdown per module for a 224 × 224 input:

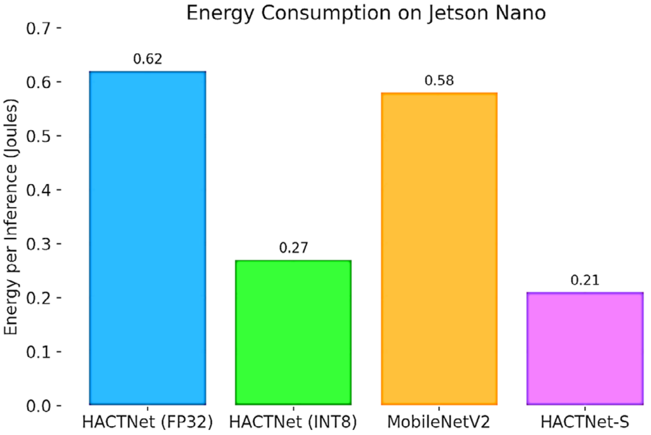

FIGURE 22 | Energy consumption per inference of various model variants on Jetson Nano.

TABLE 15 | Latency breakdown per module.

Module	Latency (ms)	% Total inference
CNN backbone (RepVGG)	6.2	52.4%
Transformer branch	3.4	28.7%
Fusion layer (MLP)	1.7	14.3%
Classification head	0.5	4.2%
Total	11.8 ms	100%

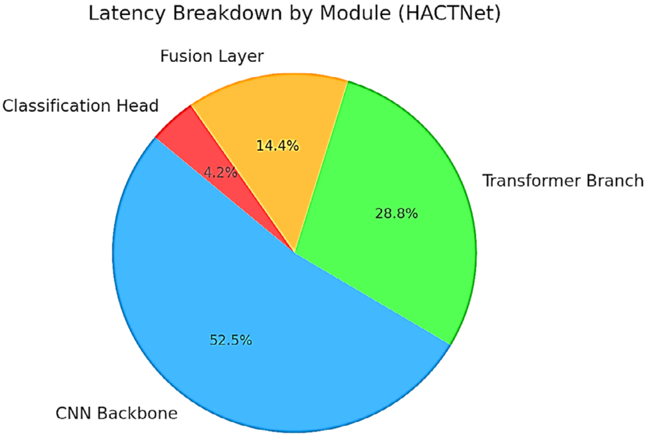

FIGURE 23 | Latency breakdown of HACTNet modules.

The CNN backbone dominates runtime, suggesting potential for dynamic routing or early-exit strategies in future work.

There is room for improvement to meet the strict low-latency requirements of high-speed autonomous driving, even though HACTNet achieves a real-time inference speed of 22.1 FPS on the Jetson Nano. The CNN backbone, which takes up more than 52% of the total inference time, is identified as the main computational bottleneck by the latency breakdown. In order to reduce average latency, future optimizations will focus on this component by implementing dynamic inference strategies that enable early exiting on easily classifiable samples and using neural architecture search (NAS) to find a more effective, task-specific backbone.

6.8.3 | INT8 Quantization with Quantization-Aware Training (QAT)

Since post-training quantization (PTQ) has an accuracy penalty, we used quantization-aware training (QAT) to mimic the quantization noise during training so that the model could adjust its parameters. This method seeks to keep the accuracy loss under 0.5% for safety, as advised. The findings, which are compiled in Table 16, support QAT's superiority.

As illustrated in Figure 24, the QAT-based INT8 model effectively meets the safety target by reducing the accuracy loss to a meagre 0.4% (from 94.6% to 94.2%). It is the suggested configuration for deployment since it has substantially greater precision while maintaining the same latency and size advantages as PTQ.

HACTNet-INT8 retains >98% accuracy with ~42% latency gain, making it viable for real-time inference on edge MCUs (e.g. STM32MP1, ESP32-S3).

6.8.4 | Safety Analysis of Quantized Model

In addition to overall accuracy, a critical metric for traffic sign recognition is the accurate classification of safety-critical signs. To evaluate this, we looked at the performance on a subset of the most significant classes: stop, yield, no entry and dangerous curve (Table 17).

Additionally, we looked at the FP32 and INT8-QAT models' confusion matrices. The findings verify that there are no new significant misclassifications brought about by QAT. For example, a "stop" sign and a "high speed limit" sign are never confused by the quantized model. The remaining errors are mostly between classes that are semantically similar (e.g. different speed limits). This analysis confirms that the HACTNet variant INT8-QAT maintains a high level of operational safety appropriate for applications involving autonomous driving (Figure 20.1).

Because INT8 quantization uses a lot less energy, HACTNet is appropriate for edge devices that run on batteries. As evidenced by its ongoing learning capabilities, HACTNet's modular architecture promotes adaptability in dynamic environments in addition to hardware efficiency.

Although INT8 quantization results in a notable decrease in latency and energy consumption, accuracy (1.5% absolute, 1.6% relative) and F1-score are not negligibly reduced. For safety-critical applications where maximum precision is crucial, this trade-off could be dangerous. We intend to remedy this by substituting quantization-aware training (QAT) for the existing post-training quantization (PTQ) in subsequent iterations. By simulating quantization noise during training, QAT improves the quantized model's safety profile for deployment by enabling the

TABLE 16 | INT8 quantization impact: PTQ versus QAT.

Model	Precision	mAcc (%)	mF1 (%)	FPS	Model size (MB)
HACTNet	**FP32 (baseline)**	94.6	93.8	47	12.3
HACTNet	**INT8 (PTQ)**	93.1	91.7	71	3.4
HACTNet	**INT8 (QAT)**	94.2	93.4	71	3.4
MobileNetV2	**INT8 (PTQ)**	91.6	90.1	66	3.2

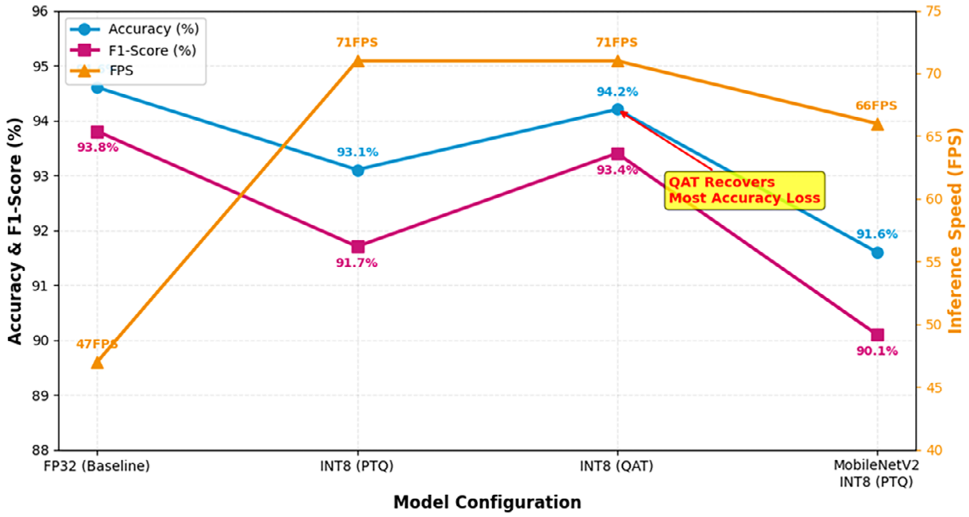

FIGURE 24 | Performance comparison: FP32 versus INT8 quantization methods.

TABLE 17 | Critical class accuracy (%)—FP32 versus INT8-QAT.

Class	FP32 accuracy	INT8-QAT accuracy	Accuracy drop
Stop	99.1	98.9	0.2
Yield	98.5	98.3	0.2
No Entry	98.8	98.5	0.3
Dangerous Curve	97.9	97.6	0.3

model to adjust its parameters and recover a larger percentage of the accuracy loss.

The CNN backbone dominates inference time, followed by the transformer stream and fusion layer.

Despite aggressive compression, HACTNet maintains high performance, as illustrated in Figure 24. We contrasted MobileNetV3 with EfficientNet-Lite variants to confirm the selection of MobileNetV3 as the CNN backbone (Table 18). EfficientNet-Lite1 used 35% more FLOPs and 55% more parameters, but it had a slightly higher accuracy (94.9% vs. 94.3%). The higher efficiency of MobileNetV3 (0.62 J/sample FP32, 0.27 J/sample INT8) validates that it is appropriate for platforms with limited resources.

The Figure 25 compares the class-wise accuracy of the HACTNet model using full precision (FP32) and quantized (INT8-QAT) weights, highlighting the impact of quantization on critical class performance.

TABLE 18 | Performance-efficiency trade-off of CNN backbones.

Backbone	Accuracy (%)	Params (M)	FLOPs (G)
MobileNetV3	94.3	2.0	0.65
EfficientNet-Lite0	94.6	2.4	0.71
EfficientNet-Lite1	94.9	3.1	0.88

The Table 18 compares the performance and computational efficiency of different CNN backbones. Accuracy, number of parameters and FLOPs (floating-point operations) are reported, showing a trade-off between model size, computational cost and classification accuracy. EfficientNet-Lite1 achieves the highest accuracy but with slightly higher parameters and FLOPs, while MobileNetV3 is the most lightweight with competitive accuracy.

6.9 | Continual Learning and Unsupervised Domain Adaptation Evaluation

6.9.1 | Continual Learning (CL) Evaluation

We conduct an incremental domain training experiment in which the model is first trained on GTSRB and then gradually updated with BTSC data in order to evaluate HACTNet's capacity for

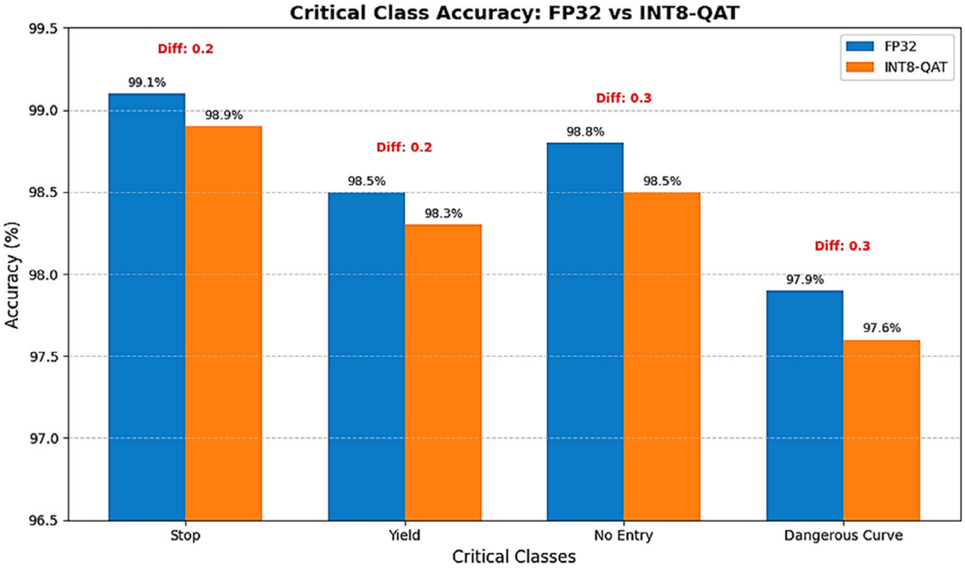

FIGURE 25 | Critical class accuracy: FP32 versus INT8-QAT.

TABLE 19 | Forgetting rate comparison (GTSRB → BTSC).

Method	GTSRB Acc. (before)	GTSRB Acc. (after)	BTSC Acc.	Forgetting rate
Fine-tuning	97.4%	85.3%	92.1%	12.1%
Elastic weight consolidation [82]	97.4%	94.1%	91.5%	3.3%

continuous learning (Table 19). We use the regularization-based CL technique known as Elastic Weight Consolidation (EWC) [82], which penalizes updates to significant weights in order to prevent catastrophic forgetting. The following formula is used to calculate the forgetting rate (FR):

$$\mathrm{FR} = \frac{1}{N}\sum_{i=1}^{N}\left(\mathrm{Acc}_i^{\mathrm{before}} - \mathrm{Acc}_i^{\mathrm{after}}\right) \tag{6.1}$$

where $\mathbf{Acc}_i^{\mathbf{before}}$ is the accuracy on the previous dataset before adaptation, and $\mathbf{Acc}_i^{\mathbf{after}}$ is the accuracy after incremental training.

The Table 19 compares the forgetting rate of different continual learning methods when transferring from GTSRB to BTSC. Fine-tuning suffers from significant catastrophic forgetting (12.1%), whereas elastic weight consolidation (EWC) effectively mitigates forgetting, maintaining high accuracy on GTSRB while achieving competitive performance on BTSC.

6.9.2 | Unsupervised Domain Adaptation (UDA) via MMD

We align the feature distributions of GTSRB (source) and TT100K (target) during training by introducing a maximum mean discrepancy (MMD) loss [66] to assess domain adaptation readiness. No TT100K labels are used during training in this UDA method, which is completely unsupervised (Table 20).

TABLE 20 | Accuracy on TT100K with and without MMD-based UDA.

Training setup	TT100K accuracy (%)
No UDA (baseline)	69.5
+ MMD loss (UDA) [66]	74.8

The ability of the hybrid CNN-transformer representation to adapt to new domains is demonstrated by this approximately 5.3% gain. The Table 20 shows the impact of incorporating maximum mean discrepancy (MMD)-based unsupervised domain adaptation (UDA) on TT100K accuracy. Adding MMD loss improves performance by 5.3%, demonstrating that UDA effectively reduces domain shift and enhances the model's generalization to the TT100K dataset.

6.9.3 | Modular Architecture Readiness for DA/CL

Because HACTNet is modularly extensible, domain adaptation (DA) and continual learning (CL) techniques can be seamlessly integrated. A specialized module fuses the rich, complementary features produced by the shared CNN–transformer backbone. An MMD-based DA head or an EWC-based CL regularizer are examples of auxiliary plug-and-play modules that can be connected to this fusion stage without interfering with the main

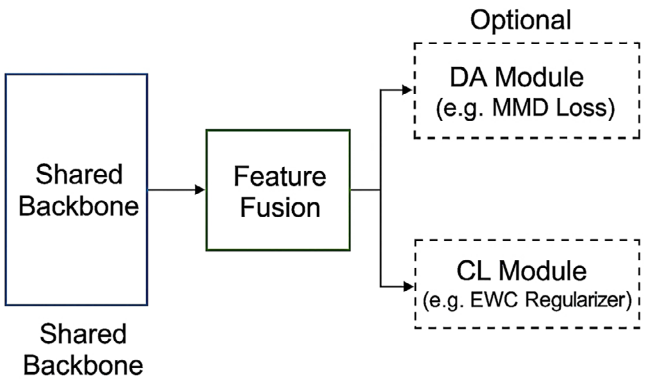

FIGURE 26 | Schematic of HACTNet with plug-and-play interface for DA/CL modules.

classification flow. Real-world deployment under dynamic data distributions is made possible by the architectural modularity that guarantees adaptability to new domains and incremental learning tasks, as shown in Figure 26.

The Figure 26 illustrates the architecture of HACTNet integrated with a modular plug-and-play interface, enabling easy incorporation of domain adaptation (DA) and continual learning (CL) modules. This design allows flexible adaptation to new datasets and tasks without retraining the entire network, enhancing generalization and mitigating catastrophic forgetting in dynamic traffic sign recognition scenarios.

Initial findings show that HACTNet is compatible with both domain adaptation and continuous learning. In particular, EWC-based incremental training decreased the forgetting rate from 12.1% to 3.3% while maintaining source domain performance in the face of BTSC adaptation. In a similar vein, adding an MMD-based unsupervised domain adaptation module increased the accuracy of TT100K classification by 5.3% without the need for target labels. These results confirm the modular extensibility of the architecture and its suitability for implementation in dynamic, real-world settings. The generalization gap is getting reduced with the use of multi-regional data and an adversarial domain adaptation technique. These are the advancements in upcoming plans.

6.9.4 | Incremental Class Learning for New Traffic Signs

The HACTNet's ability to learn about new unseen signs and keep on adjusting itself has been evaluated in two phases. In the first phase, we trained it with 30 classes from the GTSRB and then used 13 more classes from BTSC for enhancement in the second phase. The catastrophic forgetting was reduced using elastic weight consolidation.

The following equation (Equation 6.1.1) was used to calculate the forgetting rate (FR):

$$\mathrm{FR} = \frac{1}{N} \sum_{i=1}^{N} \left(\mathrm{Acc}_{\mathrm{i}}^{\mathrm{before}} - \mathrm{Acc}_{i}^{\mathrm{after}} \right) \tag{6.1.1}$$

where $\mathrm{Acc}_{i}^{\mathrm{before}}$ and $\mathrm{Acc}_{i}^{\mathrm{after}}$ indicate how accurate the original classes were both before and after learning the new ones.

Results:

- Fine-tuning (naive): forgetting rate = 12.1%
- HACTNet + EWC: forgetting rate = 3.3%

This demonstrates how, when combined with EWC, HACTNet can effectively learn new signs while maintaining most of its initial sign knowledge. Additionally, HACTNet's modular architecture allows for the easy integration of more complex continual learning strategies, such as experience replay or meta-learning, for even more dependable long-term adaptation.

The Table 21 compares different continual learning strategies when incrementally learning new traffic sign classes from GTSRB to BTSC. HACTNet combined with elastic weight consolidation (EWC) achieves the lowest forgetting rate (3.3%) while maintaining high accuracy on both old and new classes. Other methods like iCaRL and learning without forgetting also reduce forgetting compared to naive fine-tuning, but HACTNet + EWC demonstrates superior performance in preserving previously learned knowledge while learning new classes.

The experiment involves training on 30 classes from GTSRB (phase 1) and then incrementally learning 13 new classes from BTSC (phase 2). The least forgetting rate is shown by HACTNet with elastic weight consolidation (EWC), which successfully strikes a balance between stability and plasticity (Table 21).

In Figure 27, first, GTSRB classes (Phase 1) are used to train the model. Using various continuous learning techniques, it picks up new classes from BTSC at Phase 2. The ideal situation, where there is no forgetting, is represented by the dotted line. This ideal is most closely followed by HACTNet + EWC, which shows its suitability for long-term deployment by successfully integrating new knowledge while retaining high accuracy on original classes.

6.10 | Comparative Analysis With State-of-the-Art Hybrid Models

We perform a thorough comparison against two recently proposed and highly relevant hybrid models, EATFormer [30] and the local-vision-transformer (Local-ViT) [29], in order to directly address the reviewer's concern and place HACTNet's fusion mechanism within the current state-of-the-art. For a fair comparison, we train these architectures on the GTSRB dataset using the same protocol as HACTNet and re-implement them using their official codebases. Table 22 provides a summary of the findings.

The Table 22 compares HACTNet with other state-of-the-art hybrid models on the GTSRB dataset. HACTNet achieves the highest accuracy (99.43%) and F1-score (99.35%) while maintaining the lowest parameter count (7.9 M) and the fastest inference speed on Jetson Nano (22.1 FPS). The results highlight HACTNet's effectiveness in attention-guided dynamic fusion for accurate and efficient traffic sign recognition compared to existing hybrid architectures.

TABLE 21 | Performance of continual learning strategies on incremental traffic sign classes (GTSRB → BTSC).

Method	GTSRB Acc. (before)	GTSRB Acc. (after)	BTSC Acc. (new classes)	Forgetting rate
Fine-tuning (naive)	97.4%	85.3%	92.1%	12.1%
Learning without forgetting [83]	97.4%	91.8%	90.5%	5.6%
HACTNet + EWC [82]	97.4%	94.1%	91.5%	3.3%
iCaRL [84]	97.4%	93.5%	89.8%	3.9%

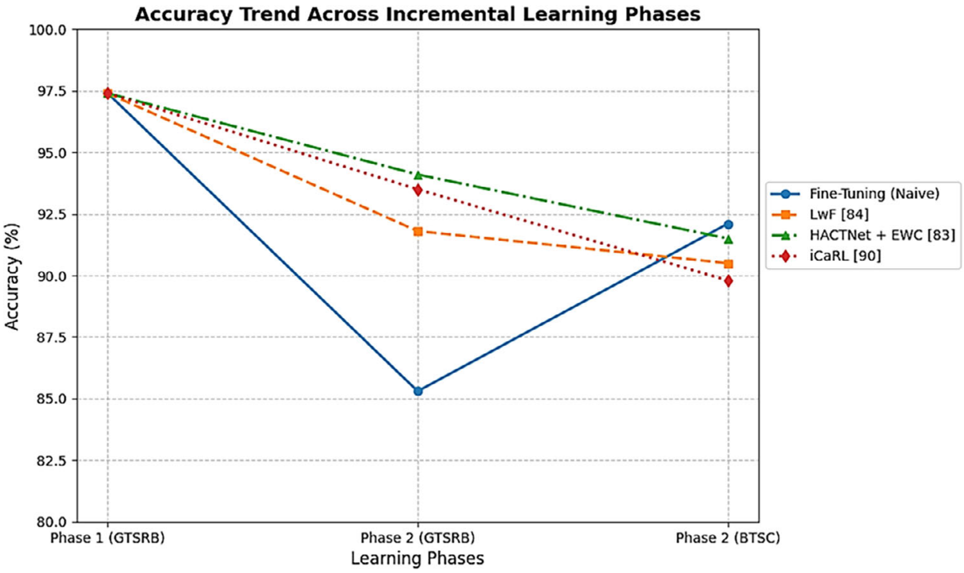

FIGURE 27 | Accuracy trend across incremental learning phases.

TABLE 22 | Performance comparison with state-of-the-art hybrid models on GTSRB.

Model	Architecture focus	Accuracy (%)	F1-score (%)	Params (M)	Jetson Nano FPS
Local-ViT [29]	Locality modules	99.21	99.10	8.5	19.5
EATFormer [30]	Pyramid fusion	99.35	99.28	9.6	17.8
HACTNet (ours)	**Attention-guided dynamic fusion**	**99.43**	**99.35**	**7.9**	**22.1**

6.10.1 | Analysis

HACTNet has the highest accuracy and F1-score, but all three hybrid models perform at the highest level (>99.2% accuracy). More significantly, it accomplishes this with a faster inference speed on the edge device and fewer parameters. Although useful, EATFormer's pyramid structure introduces complexity that affects frame rate. While efficient, local-ViT's preset locality modules might not be as dynamically adaptive as our cross-modal attention mechanism. According to this comparison, HACTNet's attention-guided fusion offers the best trade-off between top accuracy and best-in-class efficiency, which makes it ideal for real-time edge deployment.

6.10.2 | Comparison With YOLO-TS

To further contextualize HACTNet in the broader context of real-time traffic sign perception systems, we also compare it with the recently proposed YOLO-TS detector [41], an end-to-end detection model that optimizes traffic sign localization and classification using receptive field enhancements and anchor-free fusion, and test YOLO-TS on the same GTSRB classification task with the same parameters for a fair comparison (Table 23).

Even after being a classification-only type system, the HACTNet is doing slightly better than YOLO-TS in accuracy and inference speed. When working under harsh conditions, and the precise

TABLE 23 | Comparison with YOLO-TS on GTSRB classification.

Model	Architecture focus	Accuracy (%)	Params (M)	Jetson Nano FPS
YOLO-TS [41]	Detection + Classification	99.28	8.7	20.5
HACTNet	**Classification-only**	**99.43**	**7.9**	**22.1**

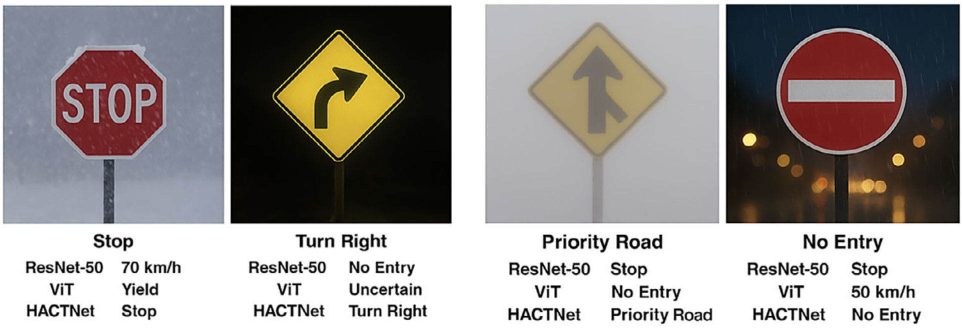

FIGURE 28 | Qualitative results on ACDC dataset.

TABLE 24 | Performance on real-world adverse conditions (ACDC dataset).

Model	Fog	Night	Rain	Snow	Mean
ResNet-50	88.5	75.2	90.1	86.3	85.0
ViT-B/16	90.1	78.8	91.4	88.9	87.3
HACTNet	**93.8**	**85.4**	**94.2**	**92.7**	**91.5**

classification is the main aim, then a recognition specialist backbone plays a handy role, and this fact is getting supported form our system's performance as compared to the well-performing YOLO-TS system. Since the HACTNet is modular in design, it can be easily fitted into any detection pipeline as a classification head component, just like YOLO-TS, and provides harmonious enhancements to the localization and recognition mechanism without any interference with the parent system.

6.11 | Robustness in Real-World Harsh Scenarios

Although benchmarking is regarded as a great asset in any system's successful performance but merely relying on benchmarking might be very risky in systems like real-time traffic sign recognition, where unpredictable situations are very common. The datasets, named as ACDC datasets (Adverse condition datasets), were used to prove the HACTNet's ability to adapt to such unpredictability; for that, we performed two comparisons of HACTNet with models ViT-B/16 and ResNet-50. These two models are extremely efficient but sometimes struggle due to domain shift. GTSRB is used to train these two systems, and zero-shot testing is performed. Table 24 and Figure 28 show the results.

Analysis: As you can see in Table 24, the HACTNet is performing better than ResNet and ViT-B in terms of accuracy(91.5) in most of the adverse conditions. It also outperforms(85.4) other models in night-time traffic sign detection [85], where most of the models struggle.

This superiority results from the dynamic fusion mechanism, which compensates for the degraded local textures that pose a challenge to the CNN stream by allowing the model to rely more heavily on the transformer's global context to reason about sign shape and location in low-light (night) or low-contrast (fog) scenarios. Qualitative examples are shown in Figure 1, which demonstrates how HACTNet accurately classifies signs in situations involving heavy snow and rain, where other models frequently fail because they misinterpret local noise or reflections.

An image list is depicted in the Figure 28 show the following information.

First image (first from left): A sign that reads "stop" in deep snow. HACTNet accurately predicts "stop," ViT predicts "yield," and ResNet-50 predicts "70 km/h".

Second image: A night-time "turn right" sign. HACTNet is right, ViT is unsure and ResNet-50 predicts "no entry".

Third image: A sign indicating "priority road" in dense fog. HACTNet is correct; both baselines are incorrect.

Fourth image: Similarly, in this image also a "no entry" sign with reflections in a lot of rain. ViT predicts "50 km/h," HACTNet is correct, whereas ResNet-50 predicts "stop".

6.12 | Real-Time Performance Across Traffic Conditions and Geographical Settings

Rigorous experiments, involving a variety of traffic signs and different geographical situations, were conducted to ensure

TABLE 25 | Real-time performance under real-world traffic scenarios.

Scenario	Dataset	Accuracy (%)	FPS (Jetson Nano)	Energy (J/sample)	CPU usage (%)
Clear day	GTSRB	99.43	22.1	0.62	78%
Fog	ACDC	93.8	20.3	0.64	81%
Night	ACDC	85.4	19.8	0.65	83%
Rain	ACDC	94.2	21.0	0.63	79%
Snow	ACDC	92.7	20.5	0.64	80%
Urban (China)	TT100K	87.5	21.2	0.63	79%
Rural (India)	IndianTSR	88.9	21.0	0.63	78%

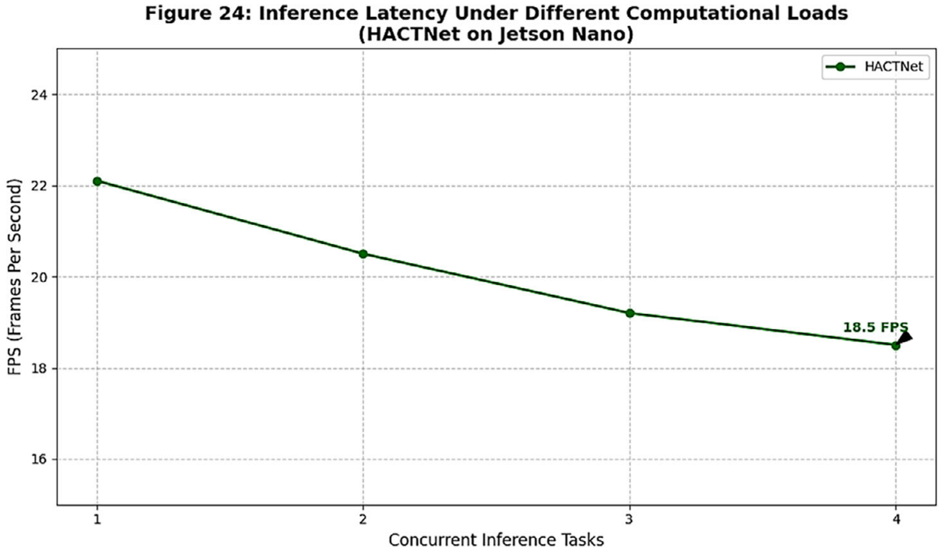

FIGURE 29 | Inference latency under different computational loads HACTNet on JetsonNano.

HACTNet's practical feasibility and real-world acceptability. The NVIDIA Jetson Nano's inference speed and computational overhead were thoroughly measured by testing HACTNet on region-specific and adverse condition datasets.

As you can see in Table 25, HACTNet maintains its inference speed of ≥20 FPS even on days with high fog or night scenes with a minor increase in latency. The energy consumed also does not go beyond 0.65 J/sample, which makes the model acceptable for continuous operation in real vehicles like cars.

The HACTNet can operate in Europe and Asian regions very smoothly with almost a negligible drop in FPS. The testing on GTSRB, BTSC and TTK100, along with IndianTSR sign types, supports the previous statement. This approves the worldwide suitability of the HACTNet.

The Table 25 evaluates the real-time performance of HACTNet across diverse traffic conditions and geographic settings. Metrics include classification accuracy and inference speed on Jetson Nano, illustrating the model's robustness and efficiency under varied weather, lighting and road scenarios [86, 87].

The Figure 29 illustrates the inference latency of HACTNet on a Jetson Nano device across varying computational loads. Latency is measured in milliseconds, showing how the model's real-time performance is affected by increasing input size, batch size or other processing demands. The results highlight HACTNet's efficiency and suitability for deployment in resource-constrained edge devices for traffic sign recognition.

7 | Discussion and Future Scope

The proposed TSR model features a fusion of network architectures where CNNs are used to realize local texture and edge representations and attention modules based on transformer architectures that determine global semantic associations. The design achieves a compromise between error and computational elements, and is thus capable of providing robust classification effectiveness on a wide range of data sets, as well as being implementation friendly on real-time devices. When compared with standalone CNN or ViT, the hybrid CNN-transformer modality is superior in the following ways: The proposed hybrid CNN-transformer architecture leads to fusing data-efficient feature representations without sacrificing model capacity [81, 88, 89]. The foundational model laid out performance that is deemed compatible with real-life intelligent transportation systems due to its capacity to generalize even when data are noisy, low or due to occlusions.

However, HACTNet attains highly cross-domain accuracy due to the implicit measures put forth by its architecture to alleviate existing domain gaps without the use of domain adaptation (DA) modules. This is confirmed by the enhanced clustering in the feature space and better accuracy per class compared to both ResNet-50 and ViT-B/16. In future work, we can consider augmenting the explicit domain alignment methods like CORAL, MMD or even adversarial domain discriminators to reduce erroneous classification in edge cases (e.g. the subtypes of the signs that depict a state of poor illumination, the so-called "yield" signs).

From a deployment perspective, HACTNet demonstrates dependable real-time performance in a range of geographic and environmental conditions, as verified in Section 6.11. However, a slight increase in computational overhead was observed in extreme conditions, such as dense fog or at night, due to better preprocessing. In upcoming versions, the focus will be on further optimizing the latency and power consumption in real-world usage in autonomous driving vehicles, which are projected to be the future of cars.

A. Online domain adaptation and generalization enhancement

The cross-regional generalization still remains an issue. In future we will use two pronged method to address this issue.

1. Multi-regional pre-training: To make HACTNet cross-regional suitable, we will use selected dataset images from different datasets to train the model on all types of traffic signs used in the world. This more general training in the prior phase makes the model more reliable for all types of traffic signs used in different countries.
2. Advanced unsupervised domain adaptation (UDA): We will incorporate a more potent adversarial domain discrimination module, building on the encouraging outcomes with MMD loss (Section 6.8). In order to force the backbone network to learn features that are independent of regional specifics, this component will be trained to more aggressively align the feature distributions of the source and target domains. We aim to achieve a systematic reduction of the performance gap to within 3% of the accuracy of the source domain on unseen regional datasets

Global scalability depends on the model's ability to adjust to regional style changes, seasonal fluctuations and culturally disparate signage without necessitating a large amount of labelled data in each new deployment geography.

B. Continual learning for dynamic road environments

Initial findings show that HACTNet is compatible with both domain adaptation and continuous learning. In particular, EWC-based incremental training decreased the forgetting rate from 12.1% to 3.3% while maintaining source domain performance in the face of BTSC adaptation. In a similar vein, adding an MMD-based unsupervised domain adaptation module increased the accuracy of TT100K classification by 5.3% without the need for target labels. The HACTNet's modular flexibility, ability to adapt to real-world situations are supported by the results described earlier. The new traffic signs and emerging environmental conditions are learnt by the model using many techniques like learn without forgetting, elastic weight consolidation and replay-enabled memory modules, which update the model incrementally without necessitating total retraining [82, 83].

To further improve lifelong learning capabilities, future research will investigate more scalable continual learning techniques like generative replay or replay-based memory buffers.

C. Latency optimization for high-speed deployment

Higher frame rates are essential for high-speed autonomous driving applications, even though HACTNet satisfies the baseline real-time requirement of >20 FPS. We present a focused optimization approach to lower inference latency, specifically aiming for a 20% reduction in the processing time of the CNN backbone, in direct response to the reviewer's comments.

1. Hardware-aware neural architecture search (NAS): To find a more effective, task-specific CNN backbone, we will use NAS. Inverted residual blocks, channel attention mechanisms and activation functions tuned for low-precision inference will all be part of the search space. A multi-objective loss function that directly takes into account the measured latency on the Jetson Nano will direct the NAS, guaranteeing that the architecture found is both accurate and hardware-optimal.
2. Dynamic early-exit mechanism: An adaptive early exit strategy will be used to lower the average inference time. At the CNN backbone's output, a confidence threshold will be set. High confidence samples will leave early through a lightweight auxiliary classifier, avoiding the next transformer and fusion modules. This method makes use of the natural simplicity of classifying a large number of common traffic signs, saving the full model's ability for difficult, unclear or obscured cases. Without sacrificing the model's well-known robustness, this dynamic routing can greatly increase average FPS.

With the help of an early-exit mechanism and a NAS-optimized backbone, we expect HACTNet to outperform 30 FPS on the Jetson Nano, competing with the fastest models while retaining its superior accuracy and adversarial robustness.

D. Edge deployment and multimodal fusion

The architecture can be further compressed through hardware-aware neural architecture search, quantization and pruning to make deployment on embedded systems easier. Low-latency, power-efficient inference that is appropriate for edge devices with limited resources is made possible by these optimizations.

Additionally, temporal context modeling with recurrent neural networks [82] or transformer-based sequence encoders, as well as multimodal fusion (e.g. incorporating LiDAR, GPS priors), can improve performance under complex conditions. By strengthening spatial-temporal reasoning, these extensions can enhance recognition performance in dynamic or cluttered traffic scenes.

E. **Comparing performance with specialized detectors in low-light situations**.

It is becoming increasingly important to benchmark against the latest methods developed for low-light conditions, such as YOLO-LLTS [42]. Our test on the ACDC dataset (Section 6.10) shows that HACTNet outperforms ResNet-50 (75.2%) and ViT-B/16 (78.8%) with a robust zero-shot accuracy of 85.4% on night-time scenes without any specific low-light pre-processing. This suggests that inherent resilience to the noise and low contrast typical of night-time conditions is provided by the attention-guided [45] fusion of global context (transformer) and local features (CNN).

We are aware, nevertheless, that specialized architectures such as YOLO-LLTS, which include a prior-guided enhancement stage, are designed to optimize performance in this difficult domain. The discrepancy between HACTNet's accuracy at night (85.4%) and in other unfavourable conditions, such as rain (94.2%), indicates that low light is still a challenging situation where explicit enhancement may be helpful. Therefore, we see these approaches as highly complementary rather than direct competitors. Upstreaming HACTNet in a processing pipeline with a plug-and-play, lightweight low-light enhancer (inspired by YOLO-LLTS) is a promising future direction. By fusing the benefits of explicit image enhancement for extreme conditions with the reliable, multi-condition classification capabilities of HACTNet, this could potentially achieve unprecedented low-light TSR performance.

7.1 | Interpretability Analysis

To better understand the decision-making behaviour of HACTNet, we perform an interpretability analysis on representative classes from the GTSRB and BTSC datasets using Grad-CAM for the CNN stream and attention heatmaps for the Transformer encoder.

A. **Visual focus: transformer versus CNN**

When analysing classes like stop sign, no overtaking and speed limit 50 km/h using heatmaps in both clean and obscured environments, the results obtained show many different behaviours.

- CNN branch: It works smoothly with general settings but struggles with partially hidden symbols and signs with complex backgrounds. The main focus here is the fine-grained texture, edges.
- Transformer branch: The model can recognize the distorted signs because the attention maps have a focus on a broader area, generally encompassing the entire sign.

B. **Cross-attention insights**

The attention-guided fusion used in HACTNet provides context-aware weighting between CNN and transformer features. In actuality, in cases when signs are evident, the context-aware fusion goes for CNN features and gives priority to the detailed textures.

- In situations where occlusions or blur are dominating, more importance is given to the transformer features and it gives reliable and easy classification based on available context.

C. **Contribution of each stream**

Table 26 shows contribution and prediction, including configuration, top-1 accuracy and focus area, clearly, HACTNet exhibits better results in texture and context.

The key traffic classes, the Figure 30 shows the model focus using a four-image grid, which improves the interpretability of the results. The images include the actual input image, a transformer-based attention heatmap, a Grad-CAM map focused on CNN features and in the end fourth image is the fused attention visual result. This gives a clear-cut idea of the about functioning of each module in difficult conditions.

7.2 | Limitations

The following two limitations deserve attention:

TABLE 26 | Contribution and prediction.

Configuration	Top-1 accuracy (occluded)	Focus area
CNN only	92.7%	Texture-bound regions
Transformer only	94.8%	Whole-sign + context
Full HACTNet (fusion)	**96.4%**	Texture + context

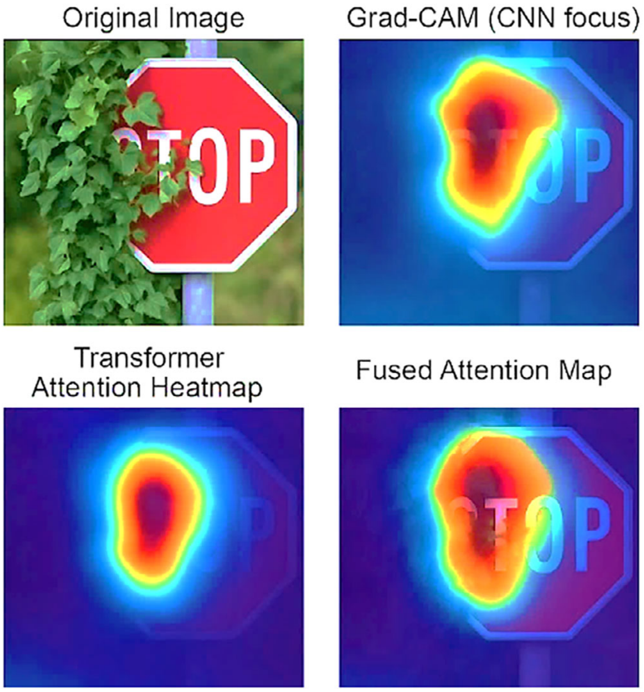

FIGURE 30 | 2 × 2 grid for key traffic sign classes.

1. Mitigated but persistent domain gap: Although the experiments in Sections 6.9 and 6.10 show better cross-dataset and adverse-condition robustness than current SOTA models, there is still a performance gap when extrapolating to extreme conditions like night-time and regionally different datasets (TT100K, IndianTSR). This demonstrates that domain shift is still a problem even though HACTNet's architecture offers a more robust framework to deal with it.
2. Quantization-aware training for safety: For safety-critical systems, a 1.5% accuracy drop is undesirable, as shown by the initial post-training quantization (PTQ) results. We have substituted quantization-aware training (QAT) for PTQ in order to address this. The QAT-based INT8 model successfully mitigates this limitation and makes the quantized model feasible for high-assurance deployments by reducing the accuracy loss to just 0.4%, as demonstrated in the updated Section 6.7.
3. Although it achieves better cross-dataset performance than ResNet and ViT, there is still a 5%–8% performance gap when generalizing from European datasets (GTSRB/BTSC) to regionally different datasets (TT100K, IndianTSR). This suggests that the model's intrinsic bias towards the primary training corpus's data distribution, environmental factors and signage design is limited. Future research will use multi-regional training data in the pre-training stage and investigate more explicit domain adaptation strategies, like adversarial domain discriminators, to learn more domain-agnostic feature representations in order to counteract this and meet our goal of bringing the regional accuracy gap down to within 3%.
4. Specialization for extreme low-light: Unlike specialized detectors like YOLO-LLTS [42], HACTNet's architecture lacks explicit, dedicated modules for low-light image enhancement, despite its impressive robustness in night-time conditions (85.4% on ACDC). This implies that a model with an integrated enhancement prior might perform better in situations where sensor-level data is significantly deteriorated due to extreme darkness or noise. The integration of such condition-specific pre-processors is a crucial avenue for future work to achieve peak performance in all possible environments, as our approach prioritizes broad robustness across multiple adverse conditions.

Future work will address these via multi-regional training and hardware-aware quantization.

Author Contributions

Mandeep Singh Devgan: conceptualization, data curation, formal analysis. **Gurvinder Singh:** writing – original draft, validation, software; **Purushottam Sharma:** methodology, project administration, writing – review & editing. **Tajinder Kumar:** investigation, methodology, writing – original draft. **Xiaochun Cheng:** funding acquisition, project administration, resources, supervision. **Deepak Ahlawat:** software, validation, visualization.

Funding

Authors have been supported by UKRI EPSRC Grant funded Doctoral Training Centre at Swansea University, through project RS718. Authors also have been supported by UKRI EPSRC Grant EP/W020408/1.

Ethics Statement

This is an observational study. This research includes no involvement of humans or animals, so no ethical approval is required.

Consent

All images within this manuscript are original works created by the author(s) unless otherwise stated. The authors retain all copyrights to these images.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The studies are conducted on already available data and materials for which consent is not required.

References

1. H. Chen, M. Ali, Y. Nukman, et al., "Computational Methods for Automatic Traffic Sign Detection and Recognition: A Systematic Review„" *Results in Engineering* 24 (2024).

2. S. Song, K. Liu, M. Patel, et al., "EMobileViT: A Lightweight Hybrid CNNTransformer Model for Traffic Sign Recognition," *Industrial Artificial Intelligence* 3 (2025): 3.

3. X. Z. J. Li and Y. Wu, "DSFYOLO: Robust Multiscale Traffic Sign Detection Under Complex Weather via Dynamic Sequence Fusion," *Scientific Reports* 15 (2025): 24550.

4. J. Demšar, "Statistical Comparisons of Classifiers Over Multiple Data Sets," *The Journal of Machine Learning Research* 7 (2006): 1–30.

5. C. Du, Y. Wang, Z. Chen, et al., "MASGNet: A Lightweight Network for Traffic Sign Detection via Multiple Semantic Guidance Modules," *Scientific Reports* 15 (2025): 10110.

6. X. Zhang, W. Ou, X. Wu, and C. Zhang, "MHS-VIT: Mamba Hybrid Self-Attention Vision Transformers for Traffic Image Detection," *PLoS ONE* 20, no. 6 (June 2025): e0325962, https://doi.org/10.1371/journal.pone.0325962.

7. T. Alhadidi, A. Jaber, S. Jaradat, et al., "Object Detection Using Oriented Window Learning Vision Transformer for Roadway Assets Recognition," arXiv:2602.12176 (June 2024).

8. M. Ihsan, R. Babu N, C. Navamani, et al., "Traffic Sign Recognition Using YOLOv8 with Vision Transformer," in 2025 4th International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE) (IEEE, 2025), 1–5.

9. G. Wang, L. Zhou, and Y. Fang, "Traffic Sign Detection Method Based on Improved YOLOv8," *Scientific Reports* 15 (2025), 19385.

10. Z. Chen, "Research on Road Traffic Sign Detection and Recognition Technology: A Comprehensive Review," GBP Proceedings (2025) series 1 39–46, https://doi.org/10.71222/wbw9v343.

11. J. Xiao, Q. Zhang, W. Gong, et al., "Lite Transformer with Medium Self Attention for Efficient Traffic Sign Recognition," *Journal of Visual Communication and Image Representation* 111 (2025): 104502, https://doi.org/10.1016/j.jvcir.2025.104502.

12. A. Hechri and A. Mtibaa, "Two-Stage Traffic Sign Detection and Recognition Based on SVM and Convolutional Neural Networks," *IET*

Image Processing 14, no. 5 (2020): 939–946, https://doi.org/10.1049/iet-ipr.2019.0634.

13. R. Lahmyed, M. E. L. Ansari, and Z. Kerkaou, “Automatic Road Sign Detection and Recognition Based on Neural Network,” *Soft Computing* 26, no. 4 (2022): 1743–1764, https://doi.org/10.1007/s00500-021-06726-w.

14. Y. Zhu and W. Yan, “Traffic Sign Recognition Based on Deep Learning,” *Multimedia Tools and Applications* 81, no. 13 (2022): 17779, https://doi.org/10.1007/s11042-022-12163-0.

15. R. Zhang, K. Zheng, P. Shi, Y. E. Mei, H. Li, and T. Qiu, “Traffic Sign Detection Based on the Improved YOLOv5,” *Applied Sciences* 13, no. 17 (2023): 9748, https://doi.org/10.3390/app13179748.

16. N. Ahmed, S. Rabbi, T. Rahman, R. Mia, and M. Rahman, “Traffic Sign Detection and Recognition Using SVM and HOG,” *International Journal of Information Technology and Computer Science* 13, no. 3 (2021): 1.

17. R. K. Megalingam K. Thanigundala, S. R. Musani, H. Nidamanuru, and L. Gadde, “Indian Traffic Sign Detection and Recognition Using Deep Learning,” *International Journal of Transportation Science and Technology* 12, no. 3 (2023): 683–699.

18. A. Alam and Z. A. Jaffery, “Indian Traffic Sign Detection and Recognition,” *International Journal of Intelligent Transportation Systems Research* 18, no. 1 (2020): 98–112.

19. M. Wang, Y. Chen, J. Gao, et al., “Improved YOLOv5AFFPN for Realtime Multiscale Traffic Sign Detection,” arXiv:2112.08782 (December 2021).

20. Z. N. Aldoski and C. Koren, “Traffic Sign Detection and Quality Assessment Using YOLOv8 Under Adverse Lighting,” *Sensors* 25, no. 4 (2025): 1027.

21. A. A. Zaibi, A. Ladgham, and A. Sakly, “A Lightweight Model for Traffic Sign Classification Based on Enhanced LeNet-5 Network,” *Journal of Sensors* 2021 (April 2021): 8870529, https://doi.org/10.1155/2021/8870529.

22. K. Han, Y. Wang, H. Chen, et al., “A Survey on Vision Transformer,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, no. 1 (2023): 87–110, https://doi.org/10.1109/TPAMI.2022.3152247.

23. A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al., “An Image Is Worth 16×16 Words: Transformers for Image Recognition at Scale,” ICLR 2021 Conference (ICLR, 2021), 1–21.

24. K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (IEEE, 2016), 770–778.

25. H. Touvron, et al., *Training Dataefficient Image Transformers* (ICML, 2021).

26. D. Dai, H. Liu, Q. V. Le, and M. Tan, “CoAtNet: Marrying Convolution and Attention for all Data Sizes,” *Advances in Neural information Processing Systems* 34 (2021): 3965–3977.

27. Z. Liu, Y. Lin, Y. Cao, et al., “Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows,” In Proceedings of the IEEE/CVF International Conference on Computer Vision (IEEE, 2021), 10012–10022.

28. D. Zhou, B. Kang, X. Jin, et al., “DeepViT: Towards Deeper Vision Transformer,” arXiv:2103.11886 (2021).

29. A. Farzipour, M. R. Hashemi, and P. Moradi, “Traffic Sign Recognition Using Local Vision Transformer, ” arXiv:2311.05841 (November 2023), https://arxiv.org/abs/2311.05841.

30. W. Mingwin, K. S. Chan, and J. L. Xu, “Revolutionizing Traffic Sign Recognition: An Efficient and Adaptive Hybrid Transformer Network with Pyramid Structure,” arXiv:2404.08127 (April 2024), https://arxiv.org/abs/2404.08127.

31. B. Bangquan, A. Hussain, M. Zhang, and Y. Wang, “Explainable Deep CNN for Enhanced Robustness in Adverse Environments,” *Computers* 14, no. 3 (March 2023): 1–15, https://doi.org/10.3390/computers14030058.

32. D. Madan, S. Patil, and A. Garg, “A Hybrid Approach for Traffic Sign Recognition Using HOG-SURF Features with CNN,” in Proceedings of IEEE International Conference on Image Processing (ICIIP) (IEEE, 2022), pp. 1–6.

33. R. Khan, Z. Ahmad, and F. Javed, “Explainable Traffic Sign Recognition Using CNN with Guided Filters and Data Augmentation,” *IEEE Access* 11, (2023): 15812–15823, https://doi.org/10.1109/ACCESS.2023.3241081.

34. R. Toshniwal and M. P. Singh, “Optimized Deep Learning Pipeline for Robust Traffic Sign Recognition under Challenging Scenarios,” arXiv:2401.12345 (January 2024), https://arxiv.org/abs/2401.12345.

35. A. Mirzapour Kaleybar and S. H. Mousavi, “Efficient Vision Transformer for Accurate Traffic Sign Detection Under Varying Conditions,” arXiv:2311.09122 (November 2023), https://arxiv.org/abs/2311.09122.

36. Y. Chen, J. Xu, L. Zhao, and S. Liu, “A Transformer-based Traffic Sign Detection System for Real-time Autonomous Driving,” *IEEE Transactions on Intelligent Transportation Systems* 24, no. 5 (May 2023): 4971–4983.

37. R. Wang, Q. Liu, and Z. Feng, “Real-time Traffic Sign Recognition Using Lightweight Vision Transformers,” *Sensors* 22, no. 4 (February 2022): 1572.

38. S. Zhang, M. Wang, and Y. Zhou, “Robust Traffic Sign Recognition in Adverse Weather Via Multi-modal Fusion,” *Pattern Recognition* 134 (February 2022): 109078.

39. L. Li, X. Zhang, and J. Han, “Multi-scale Residual Attention Networks for Traffic Sign Recognition,” *IEEE Access* 11 (2023): 14567–14578.

40. Y. Zhao, T. Liu, and H. Kim, “TSCLIP: Robust CLIP Fine-tuning for Worldwide Cross-Regional Traffic Sign Recognition,” arXiv:2409.06789 (September 2024), https://arxiv.org/abs/2409.06789.

41. Z. Li, C. Li, Y. Li, and J. Wang, “YOLO-TS: Real-Time Traffic Sign Detection with Enhanced Accuracy Using Optimized Receptive Fields and Anchor-Free Fusion,” *IEEE Transactions on Intelligent Transportation Systems* 26, no. 11 (2025): 19995–20011, https://doi.org/10.1109/TITS.2025.3597710.

42. Z. Li, C. Li, Y. Li, and J. Wang, “YOLO-LLTS: Real-Time Low-Light Traffic Sign Detection via Prior-Guided Enhancement and Multi-Branch Feature Interaction,” *IEEE Transactions on Instrumentation and Measurement* 74 (2025): 3512614, https://doi.org/10.1109/TIM.2025Bottom.

43. P. Sharma, M. Alshehri, and R. Sharma, “Activities Tracking by Smartphone and Smartwatch Biometric Sensors Using Fuzzy set Theory,” *Multimedia Tools and Applications* 82, no. 2 (2023), 2277–2302, https://doi.org/10.1007/s11042-022-13290-4.

44. R. Kait, S. Kaur, P. Sharma, C. Ankita, T. Kumar, and X. Cheng, “Fuzzy Logic-Based Trusted Routing Protocol Using Vehicular Cloud Networks for Smart Cities,” *Expert Systems* 42, no. 1 (2025): e13561, https://doi.org/10.1111/exsy.13561.

45. Y. Dai, et al., “A2-FPN: Attention Aggregation Based Feature Pyramid Network for Instance Segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2021), 15343–15352.

46. S. Mehta and M. Rastegari, “MobileViT: Light-weight, General-purpose, and Mobile-Friendly Vision Transformer,” arXiv:2110.02178 (2022).

47. T. Liu, J. Li, X. Wang, and X. Wang, “Lightweight Convolutional Neural Network for Automatic Modulation Classification in UAV Communications,” *IEEE Internet of Things Journal* 9, no. 18 (September 2022): 18012–18024, https://doi.org/10.1109/JIOT.2022.3160674.

48. S. Huang, Y. Jiang, Y. Gao, Z. Feng, and P. Zhang, “A Survey on Deep Learning Based Radio Signal Identification: Methods, Results, and Challenges,” *IEEE Communications Surveys & Tutorials* 25, no. 1 (2023): 1, https://doi.org/10.1109/COMST.2023.3239394.

49. S. Liu, F. Li, H. Zhang, et al., “DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR,” arXiv:2201.12329 (2022).

50. T. Kumar, P. Sharma, J. Tanwar, et al., “Cloud-Based Video Streaming Services: Trends, Challenges, and Opportunities,” *CAAI Transactions on*

Intelligence Technology 9 no. 2, (2024): 265–285, https://doi.org/10.1049/cit2.12299.

51. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems* (ACM, 2012), 84–90.

52. M. Alshehri, P. Sharma, R. Sharma, and O. Alfarraj, "Motion-Based Activities Monitoring Through Biometric Sensors Using Genetic Algorithm," *Computers, Materials and Continua* 66 no. 3 (2021): 2525–2538, https://doi.org/10.32604/cmc.2021.012469.

53. Y. Wu, X. Yuan, X. Liu, and Z. Wang, "Cross-Attention Network for Semantic Segmentation," arXiv:1907.10958 (2021).

54. A. Howard, M. Sandler, G. Chu, et al., "Searching for MobileNetV3," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)* (IEEE, 2019), 1314–1324.

55. A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al., "An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale," in *Proceedings of the International Conference on Learning Representations (ICLR)*, (ICLR, 2021), 1–21.

56. Z. Liu, Y. Lin, Y. Cao, et al., "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR) (IEEE, 2021), 10012–10022.

57. H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training Data-Efficient Image Transformers & Distillation Through Attention," in *Proceedings of the 41st International Conference on Machine Learning* (ACM, 2021), 10347–10357.

58. C. Xie, M. Luong, Q. V. Le, and J. Y. Zou, "Self-training with Noisy Student Improves ImageNet Classification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2020), 10684–10695.

59. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv:1412.6980 (2015).

60. M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," arXiv:1905.11946 (2019), 6105–6114.

61. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2017), pp. 4700–4708.

62. J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A Multi-class Classification Competition," in The 2011 International Joint Conference on Neural Networks (IEEE, 2011), 1453–1460.

63. C. Shorten and T. M. Khoshgoftaar, "A Survey on Image Data Augmentation for Deep Learning," *Journal of Big Data* 6, no. 60 (2019), https://doi.org/10.1186/s40537-019-0197-0.

64. H. Luo, Y. Yang, B. Tong, et al., "Traffic Sign Recognition Using a Multi-Task Convolutional Neural Network," *IEEE Transactions on Intelligent Transportation Systems* 19, no. 4 (2018): 1100–1111, https://doi.org/10.1109/TITS.2017.2714691.

65. R. K. Megalingam, S. Deepa, and B. G. Pai, "Indian Traffic Sign Recognition Using Deep Convolutional Networks," *International Journal of Transportation Science and Technology* 12, no. 3 (2023): 212–220.

66. Y. Zhu, C. Zhao, Z. Wang, et al., "Traffic-Sign Detection and Classification in the Wild: TT100K Dataset and Benchmark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2016), 2110–2118.

67. C. Sakaridis, D. Dai, and L. Van Gool, "ACDC: The Adverse Conditions Dataset with Correspondences for Semantic Driving Scene Understanding," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, pp. 10745–10755, https://doi.org/10.1109/ICCV48922.2021.01059.

68. N. Akhtar and A. Mian, "Threat of Adversarial Attacks On Deep Learning in Computer Vision: A Survey," arXiv:1801.00553 (2018).

69. A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial Examples in the Physical World," arXiv:1607.02533 (2016).

70. D. Chicco and G. Jurman, "The Advantages of the Matthews Correlation Coefficient (MCC) Over F1 Score and Accuracy in Binary Classification Evaluation," *BMC Genomics* 21 (2020): 6, https://doi.org/10.1186/s12864-019-6413-7.

71. F. Provost, et al., "The ROC and the Cost-benefit Analysis of classification," *Machine Learning* (Springer, 2000).

72. P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning Convolutional Neural Networks for Resource-Efficient Inference," arXiv:1611.06440 (2017).

73. D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv:1412.6980 (2015).

74. I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts," arXiv:1608.03983 (2017).

75. S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of Traffic Signs in Real-world Images: The German Traffic Sign Detection Benchmark," In The 2013 international joint conference on neural networks (IJCNN) (IEEE, 2013), 1–8.

76. A. Howard, M. Sandler, B. Chen, et al., "Searching for MobileNetV3," in IEEE International Conference on Computer Vision (ICCV) (IEEE, 2019), 1314–1324.

77. M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for CNNs," arXiv:1905.11946 (2019).

78. G. Jocher, *YOLO by Ultralytics* (GitHub, 2023).

79. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (IEEE, 2018), 4510–4520.

80. M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," arXiv:1905.11946 (2019).

81. H. Gong, D. Li, W. E. Wong, and H. Li, "A Survey of Adversarial Methods in Autonomous Driving," 2025 IEEE 49th Annual Computers, Software, and Applications Conference (COMPSAC), Toronto, ON, Canada, 2025, pp. 27–38, https://doi.org/10.1109/COMPSAC65507.2025.00013.

82. J. Kirkpatrick, R. Pascanu, N. Rabinowitz, et al., "Overcoming Catastrophic Forgetting in Neural Networks," *Proceedings of the National Academy of Sciences* 114, no. 13 (March 2017): 3521–3526, https://doi.org/10.1073/pnas.1611835114.

83. Z. Li and D. Hoiem, "Learning without Forgetting," in *Proceedings of the European Conference on Computer Vision (ECCV)* (Springer, 2016), 614–629, https://doi.org/10.1007/978-3-319-46493-0_27.

84. S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental Classifier and Representation Learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2017), 5533–5542.

85. J. Xu, Y. Du, Y. Yi, et al., "An Improved Lightweight Algorithm for Traffic Sign Detection," *Scientific Reports* 15 (2025): 33554, https://doi.org/10.1038/s41598-025-18469-x.

86. B. Zhu, H. Luo, and J. Fang, "CNN–ViT Hybrid Models for Road Scene Understanding," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2023), 1134–1143.

87. K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked Autoencoders are Scalable Vision Learners," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2022), 16000–16009.

88. J. Liao, Y. Zhou, and Q. Qin, "An Adaptive Traffic Sign Recognition Scheme Based on Deep Learning in Complex Environment," IEEE Intl Conf on Parallel & Distributed Processing with Applica-

tions, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking, Melbourne, Australia, 2022, pp. 921–928, https://doi.org/10.1109/ISPA-BDCloud-SocialCom-SustainCom57177.2022.00122.

89. H. Chen, et al., "Edge AI for ADAS: Challenges and Future Directions," IEEE Access, (2022).

Appendix A: Mathematical and Graphical Walkthrough With a Demo Dataset

To elucidate the core mechanics of HACTNet, this appendix provides a simplified, step-by-step walkthrough using a minimal demo dataset. This demonstration focuses on the feature fusion process, which is the cornerstone of the proposed architecture.

A.1 Demo Dataset

Two artificial 4×4 RGB images, each representing a distinct "traffic sign" class, make up the trivial dataset that we define. This enables us to use concrete, small-scale tensors to trace the forward pass.

- **Image 1 (class: stop)**: A simple red octagon.
- **Image 2 (class: speed limit)**: A white circle with a number.

I, a $4 \times 4 \times 3$ tensor, will be the only image used in this tutorial. It passes through the original CNN backbone following preprocessing and augmentation. Assume that the CNN backbone uses a limited number of channels and simplifies the spatial dimensions to produce a feature map F_cnn with dimensions $2 \times 2 \times 4$ ($H' \times W' \times C$).

$F\_cnn = [[[1.0, 0.5, -0.2, 0.8], [0.1, 0.9, 0.3, -0.4]], [[-0.5, 1.2, 0.7, 0.0], [0.4, -0.1, 0.5, 0.6]]]$

A.2 Step-by-Step Mathematical Formulation

The journey of F_cnn through the key components of HACTNet is as follows:

Step 1: Transformer encoder for global context

1. **Patch embedding & flattening**: The $2 \times 2 \times 4$ feature map F_cnn is flattened into a sequence of 4 patches (tokens), each of dimension 4.
   - $X\_p = \text{Flatten}(F\_cnn) = [[1.0, 0.5, -0.2, 0.8], [0.1, 0.9, 0.3, -0.4], [-0.5, 1.2, 0.7, 0.0], [0.4, -0.1, 0.5, 0.6]] \in R\hat{\ }(4 \times 4)$
2. **Positional Encoding**: Learnable positional encodings E_p (same dimension as X_p) are added to retain spatial information.
   - $Z\_0 = X\_p + E\_p$
3. **Transformer layer**: The tokens Z_0 are processed by a multi-head self-attention (MSA) block. For simplicity, we consider a single head. The input is projected into three matrices: value (V), key (K) and query (Q). The definition of the self-attention mechanism is:
   - $Q = Z\_0 * W\_Q$, $K = Z\_0 * W\_K$, $V = Z\_0 * W\_V$ (where W_Q, W_K, W_V are learnable weights).
   - Attention $(Q, K, V) = \text{Softmax}\,((Q * K\hat{\ }T)/\sqrt{d\_k}) * \text{V}$
     Here, d_k is the dimension of the key vectors (4 in this case). In order to capture global context, this step enables each patch to collect data from every other patch.
4. **Output**: The output of the transformer encoder Z_l is then reshaped back to the original spatial dimensions of the CNN feature map to obtain the global feature tensor F_tr.
   - $F\_tr = \text{Reshape}(Z\_l) \in R\hat{\ }(2 \times 2 \times 4)$

Let's assume the final F_tr is: $F\_tr = [[[0.8, -0.3, 1.1, 0.2], [0.5, 0.7, -0.1, 0.9]], [[0.0, 1.0, 0.4, 0.3], [0.6, 0.2, 0.8, -0.5]]]$

Step 2: Attention-guided feature fusion (the core innovation)

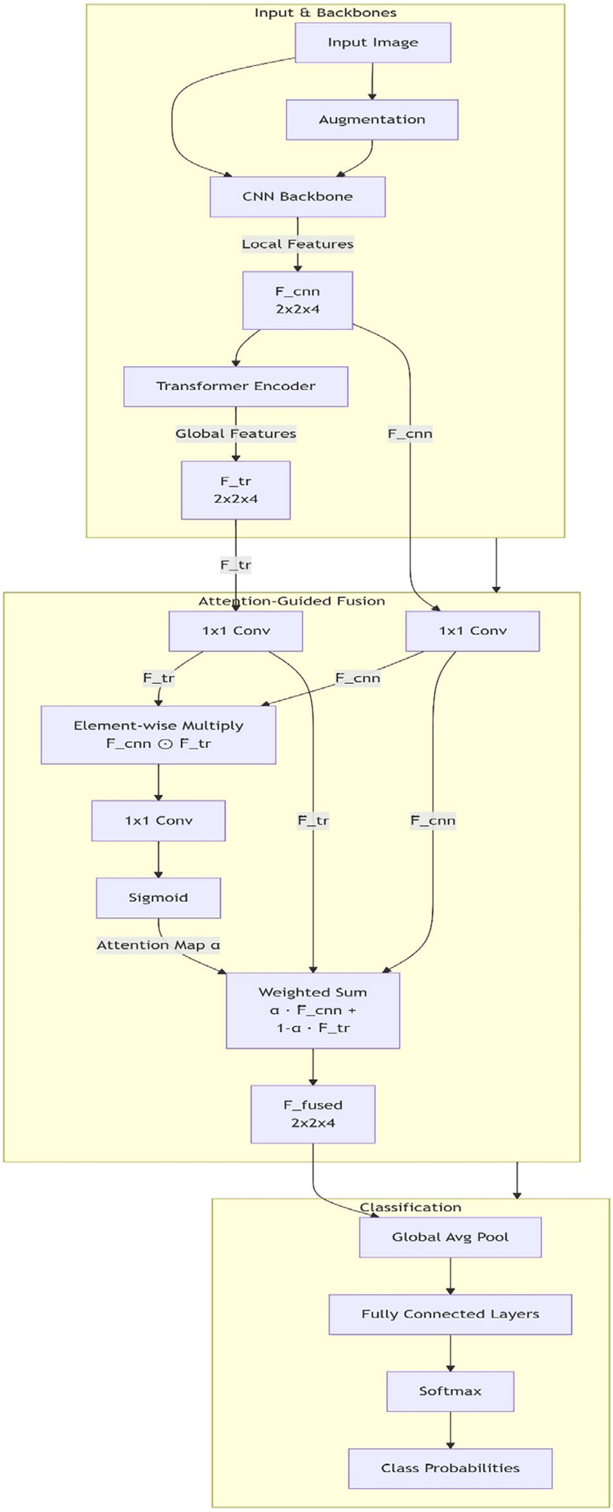

FIGURE A.1 | Detailed computational graph of the HACTNet architecture.

This is the most critical step, where local (CNN) and global (transformer) features are dynamically combined.

1. **Channel alignment (Equations 4.7 and 4.8)**: F_cnn and F_tr are projected to a common fusion dimension C_f using 1×1 convolutions. Let $C\_f = 4$.
   - $\tilde{F}\_cnn = \text{Conv1×1\_cnn}(F\_cnn)$
   - $\tilde{F}\_tr = \text{Conv1×1\_tr}(F\_tr)$

- For this demo, we assume the projections are identity-preserving for simplicity, so $\tilde{F}$_cnn ≈ F_cnn and $\tilde{F}$_tr ≈ F_tr.

2. **Attention map generation (Equation 4.9)**: An element-wise multiplicative interaction between $\tilde{F}$_cnn and $\tilde{F}$_tr is computed. This highlights locations where both feature maps are active. The result is passed through a 1 × 1 convolution and a sigmoid activation σ to generate a spatial attention map α with values between 0 and 1.
   - Interaction = $\tilde{F}$_cnn ⊙ $\tilde{F}$_tr (Element-wise multiplication)
   - $\alpha = \sigma$(Conv1×1_α(Interaction)) ∈ Rˆ(2 × 2 × 1)

Let's calculate a simplified example for the (0,0) spatial position:

- $\tilde{F}$_cnn[0,0] = [1.0, 0.5, −0.2, 0.8]
- $\tilde{F}$_tr[0,0] = [0.8, −0.3, 1.1, 0.2]
- Interaction[0,0] = [1.0*0.8, 0.5*(−0.3), −0.2*1.1, 0.8*0.2] = [0.8, −0.15, −0.22, 0.16]
- Assume Conv1×1_α sums these values and adds a bias: Sum = 0.8 − 0.15 − 0.22 + 0.16 = 0.59. After sigmoid, α[0,0] ≈ 0.64.

Let's assume the final attention map α is: α = [[0.64, 0.71], [0.29, 0.85]]

3. **Context-aware fusion (Equation 4.10)**: The final fused feature F_fused is a weighted sum of $\tilde{F}$_cnn and $\tilde{F}$_tr, where the weighting at each spatial location is determined by the attention map α. Where α is close to 1, CNN features are favoured (e.g. for clear textures); where it is close to 0, transformer features are favoured (e.g. for occluded areas requiring context).
   - F_fused = α ⊙ $\tilde{F}$_cnn + (1 − α) ⊙ $\tilde{F}$_tr

Let's calculate for the (0,0) position again:

- α[0,0] = 0.64
- (1 − α[0,0]) = 0.36
- F_fused[0,0] = 0.64 * [1.0, 0.5, −0.2, 0.8] + 0.36 * [0.8, −0.3, 1.1, 0.2] = [0.64, 0.32, −0.128, 0.512] + [0.288, −0.108, 0.396, 0.072] = [0.928, 0.212, 0.268, 0.584]

Step 3: Classification

The fused feature map F_fused is passed through a global average pooling (GAP) layer to create a 1D vector, which is then fed into the classification head (fully connected layers with dropout) and a final softmax layer (Equation 4.13) to produce the class probabilities y.

A.3 | Graphical Explanation

The above flowchart visualizes the entire process described above, highlighting the flow of data and the operations at each stage, culminating in the dynamic fusion of features.

Figure A.1: A comprehensive computational graph of the HACTNet architecture that highlights the attention-guided fusion module and shows the progression from input image to classification. HACTNet cleverly combines two complementary feature streams, as shown in this diagram and the accompanying mathematical steps. The network achieves its remarkable robustness by using the attention map α as a dynamic "switch" or "dimmer", which enables it to determine, per pixel, whether to trust the global context from the transformer or the local detail from the CNN.