Dean&Francis
ISSN 2959-6157

Design of an Embedded Lightweight System for Traffic Signal Detection in Adverse Weather and at Night Based on YOLOX-Nano

Hongru Zheng

Department of Electronics, Guangdong Ocean University, Guangzhou, 524000, China
13538970444@stu.gdou.edu.cn

Abstract:

With the rapid advancement of intelligent transportation systems (ITS) and autonomous driving technologies, the robustness of traffic signal detection under complex meteorological and nighttime conditions has emerged as a critical concern. To address challenges such as diminished recognition accuracy in adverse weather and low-light conditions, as well as deployment constraints on edge devices, this paper proposes an embedded lightweight traffic signal detection system based on YOLOX-Nano. The system enhances environmental adaptability via a diversified meteorological scene data augmentation strategy, while achieving model lightweighting through network pruning and mixed-precision quantization, effectively balancing detection accuracy, real-time performance, and resource efficiency within the constraints of limited hardware resources [1]. During the deployment phase, the system utilizes ONNX and TensorRT to establish an efficient inference toolchain, enabling cross-framework model conversion and acceleration optimization, ultimately achieving low-power, high-performance real-time traffic signal detection on the Jetson Nano platform. The primary focus of this paper is on the system architecture design and its implementation pathway, aiming to provide an embedded lightweight solution with engineering feasibility and scalability for traffic signal detection under complex meteorological conditions.

Keywords: YOLOX-Nano; embedded system; Jetson Nano; TensorRT; Weather Condition Monitoring

ISSN 2959-6157

1. Introduction

Traffic lights are a core visual element in road traffic management and vehicle intelligent decision-making systems. The accuracy and responsiveness of traffic signal detection directly impact driving safety and traffic operational efficiency. However, in practical applications, environmental perturbations such as raindrop occlusion, fog scattering, nighttime low illumination, and intense light reflection can significantly degrade image quality, causing traditional detection models to degrade in the feature extraction and recognition stages or even make misjudgments [2]. Concurrently, embedded computing platforms (such as Jetson Nano) are constrained by limited computational power and storage resources, making it difficult for standard deep learning models to achieve high-precision inference while ensuring real-time performance. Typical works, such as Jayasinghe et al., proposed an edge detection scheme based on SSD-MobileNetV2 and TensorRT, and achieved a real-time inference performance of 63 FPS on the Jetson Xavier platform [3]. However, the model size and computational overhead of this scheme remain ill-suited for the lower computing power and stricter power consumption constraints of Nano-level edge platforms. To address this issue, this paper designs an embedded lightweight traffic signal detection system based on YOLOX-Nano, specifically targeting severe weather and nighttime scenarios. This system introduces a lightweight optimization strategy at the model structure level and combines multi-meteorological scenario data augmentation with TensorRT inference acceleration technology to achieve efficient detection and real-time embedded deployment in complex environments.

The main contributions and structure of this paper are as follows: Chapter 2 presents the overall architecture and workflow of the system; Chapter 3 elaborates on the model design and lightweight optimization methods; Chapter 4 details the embedded deployment and inference acceleration mechanisms; Chapter 5 conducts a theoretical analysis on the system's performance and robustness; Chapter 6 summarizes the entire paper and looks forward to future research directions.

2. System Overall Architecture

2.1 System Composition

This system is primarily composed of three parts: the data augmentation module, the detection model module, and the embedded deployment module. The data augmentation module is designed to enhance the diversity and robustness of traffic signal image datasets. Through the implementation of physical augmentation operations on raw images, such as raindrop occlusion, fog scattering, low-light simulation, and reflected light interference, it establishes a training sample dataset with complex meteorological characteristics to improve the model's generalization ability under multiple scene conditions. The detection model module is based on the YOLOX-Nano network structure, incorporating depthwise separable convolutions to reduce model parameters and computational complexity. It further adopts a decoupled detection head structure to decouple classification and regression tasks, enhancing feature representation capability and inference efficiency. This module achieves a substantial reduction in computational overhead while preserving high detection precision, laying a solid model foundation for embedded deployment. The embedded deployment module uses Jetson Nano as the core hardware platform and leverages the ONNX and TensorRT toolchains to achieve model conversion, operator fusion, and inference acceleration. In response to the resource constraints of edge devices, this module accomplishes the cross-framework migration of the model from PyTorch to the TensorRT engine, empowering the system to achieve real-time traffic signal detection in a low-power consumption environment.

2.2 System Workflow

The system's operation flow comprises four stages: model training, model export, model optimization, and embedded deployment. In the model training stage, the YOLOX-Nano model is trained on a dataset enhanced with multiple weather conditions within the PyTorch framework. Supervised learning is employed to optimize network parameters, enabling the model to accurately recognize traffic signals across diverse environmental contexts. In the model export stage, the trained weight file is converted to ONNX format, achieving model framework independence and enhancing cross-platform compatibility. In the model optimization stage, TensorRT is used to optimize the computational graph, fuse operators, and quantize precision, thereby improving inference speed and resource utilization efficiency on the Jetson Nano platform, and generating a directly executable TensorRT engine file. In the embedded deployment stage, the system loads the optimized engine file onto the Jetson Nano, performs inference on real-time video streams captured by an on-board camera, and outputs the traffic light category and location, achieving real-time and stable visual recognition functionality on the edge device.

3. Model Design and Lightweight Optimization

3.1 Model structure design

YOLO series models partition the input image into an S×S grid, enabling each grid to predict multiple bounding boxes along with their corresponding confidence scores, and simultaneously output the conditional probability of the class, forming a unified end-to-end detection tensor structure [4]. Leveraging inverted residual and linear bottleneck structures, MobileNetV2 further formalizes depthwise separable convolution, significantly reducing computational overhead while preserving effective feature representation capability [5]. This system uses YOLOX-Nano as the basic framework, and its overall structure consists of three parts: feature extraction (Backbone), feature fusion (Neck), and detection output (Head). The core PEP (Projection–Expansion–Projection) module realizes efficient feature encoding via a lightweight architecture characterized by "dimensionality reduction–dimensionality increase–depth convolution–re-dimensionality reduction", effectively reducing the amount of computation while maintaining model accuracy [1]. To address the small size and significant appearance variations of traffic lights, this paper proposes targeted modifications to the original structure: Depthwise separable convolutions are introduced into the Backbone to reduce computational overhead; a lightweight path aggregation structure (PAN) is employed in the Neck layer to strengthen multi-scale feature propagation and fusion capabilities; and a decoupled detection head is used in the Head component, modeling classification and regression separately to improve gradient stability and small target detection performance.
The overall model structure is illustrated in Figure 1. Following these structural optimizations, the system achieves a significant reduction in overall computational complexity while maintaining detection accuracy, thus providing a solid structural foundation for subsequent deployment on embedded platforms.

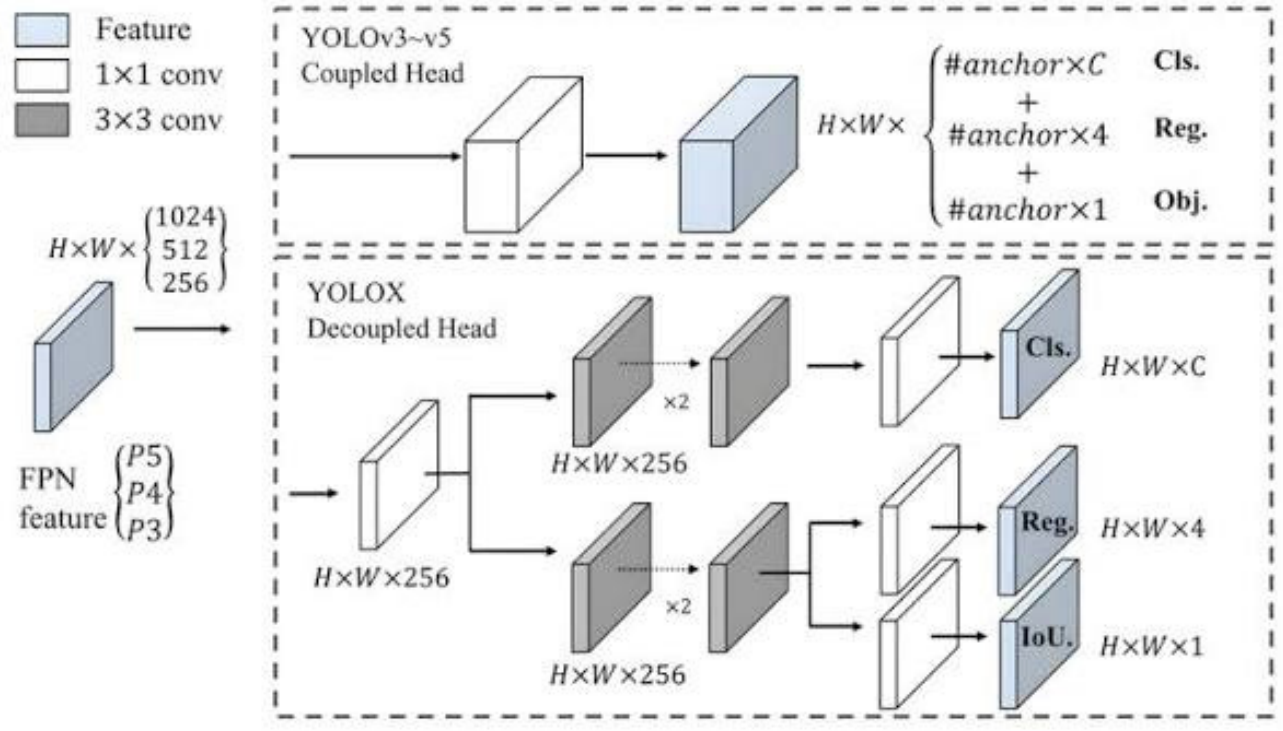

Figure 1. YOLOX-Nano architecture.

3.2 Lightweight optimization methods

FasterX proposes a lightweight PixSF Head and SlimFPN based on YOLOX, and validates the real-time detection capability for small targets on Jetson NX/Nano, highlighting the advantage of "simplifying the neck and head" on edge latency [6]. To accommodate the resource constraints of Jetson Nano, this paper introduces a joint optimization strategy combining structural pruning and mixed-precision quantization post-model training. In the pruning stage, redundant feature channels are automatically pruned according to the importance index of channel weights to reduce the model size; in the quantization stage, redundant feature channels are adaptively pruned based on the impor-

ISSN 2959-6157

tance ranking of channel weights to reduce memory usage and improve computational parallelism. The synergistic effect of pruning and quantization reduces the number of model parameters by about 40% and the floating-point operation volume (FLOPs) by about 35%, thereby achieving higher energy efficiency and real-time performance on embedded devices. The computational complexity of standard convolution can be expressed as:

$$FLOPs_{std} = H \times W \times C_{in} \times C_{out} \times k^2 \tag{1}$$

The depthwise separable convolution is decomposed into two parts: channel-wise convolution and pointwise convolution. The computational complexity of the optimized model can be expressed as:

$$\mathrm{FLOP}_{S_{dw}} = H \times W \times \left(C_{in}k^2 + C_{in}C_{out}\right) \tag{2}$$

The ratio of the two equations gives the acceleration ratio:

$$r = \frac{\mathrm{FLOPs}_{dw}}{\mathrm{FLOPs}_{std}} = \frac{1}{C_{out}} + \frac{1}{k^2} \tag{3}$$

As indicated by equation (3), the speedup ratio r is inversely proportional to both the kernel size and the number of output channels. When the number of network channels is large, depthwise separable convolution can significantly reduce the computational cost. When the kernel size is $k = 3$ and the number of output channels is $C_{out} = 128$, we have $r \approx \frac{1}{128} + \frac{1}{9} \approx 0.12$, which means the computational cost is about 12% of that of standard convolution, theoretically achieving an 88% reduction in computational overhead. Considering that some layers in the network (such as the Stem layer and the output layer) still employ conventional convolution, the overall average reduction is about 65%–70%. Additionally, after model training, structural pruning removes feature maps with minimal channel weight contributions, and the mixed-precision quantization (FP32→FP16) strategy further compresses weight scales—further reducing the overall number of parameters by about 40% and the computational complexity (FLOPs) by about 35%. The optimized model can achieve low-latency real-time inference on the Jetson Nano platform while maintaining high detection accuracy. Han et al.'s research demonstrates that the combined strategy of pruning and quantization can significantly reduce the parameter size while maintaining model accuracy, which provides a theoretical basis for the adoption of channel pruning combined with hybrid precision quantization in the embedded platform in this paper [7].

3.3 Loss function and training objective

The total system loss function is defined as:

$$\mathcal{L} = \lambda_{\mathrm{cls}}\mathcal{L}_{\mathrm{cls}} + \lambda_{\mathrm{obj}}\mathcal{L}_{\mathrm{obj}} + \lambda_{\mathrm{box}}\mathcal{L}_{\mathrm{box}} \tag{4}$$

Among them, $\mathcal{L}_{cls}$ denotes the classification loss of traffic signals, $\mathcal{L}_{obj}$ represents the objectness loss, and $\mathcal{L}_{box}$ corresponds to the bounding-box regression error. The bounding-box regression adopts the CIoU metric to jointly constrain overlap, center-point distance, and aspect-ratio consistency of the predicted box, thus improving the localization precision for small targets such as distant traffic lights. Existing studies have shown that the EIoU loss function can further stabilize the bounding-box regression process by more effectively constraining center deviation and aspect-ratio variation, exerting a positive effect on enhancing localization stability under complex weather conditions [8].

3.4 Embedded Deployment and Inference Acceleration

After model training, a PyTorch→ONNX→TensorRT model conversion and deployment pipeline was employed. First, the PyTorch model was exported to ONNX format to achieve cross-framework compatibility. Subsequently, NVIDIA TensorRT tools were utilized to implement operator fusion, computational graph simplification, and precision quantization for the model. Finally, a TensorRT engine file that can be directly loaded onto Jetson Nano was generated. Figure 2 illustrates the embedded deployment process of the model. The system performs inference on Jetson Nano using real-time video streams from a camera to detect and locate traffic lights. Theoretical analysis shows that the system has real-time processing capabilities. This section mainly describes the deployment process of the model on the embedded side; its specific performance analysis in terms of resource consumption and real-time performance will be further quantitatively discussed in Chapter 4.

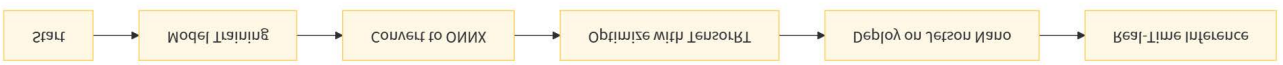

Figure 2 .Model Deployment Flowchart

4. System performance theory and feasibility analysis

4.1 Model complexity analysis

Based on the model design in Chapter 3, this paper significantly reduces the parameter size and computational cost through depthwise separable convolution and network pruning techniques. Theoretical computations indicate that the optimized model possesses approximately 1.3M parameters and a computational complexity of around 1.5 GFLOPs (input resolution: 640×640). In comparison with the conventional YOLOv5s model (approximately 7.2M parameters, ~16 GFLOPs), the parameter size is reduced by more than 60%, and the computational complexity is reduced by about 70% [9]. These results demonstrate that the lightweight structure designed in this paper can effectively reduce computational overhead while ensuring detection accuracy, laying a structural foundation for embedded real-time deployment.

4.2 Platform compatibility analysis

This section assesses the performance and resource consumption of the deployment scheme described in Section 3.4 using theoretical calculations and platform-specific metrics. The Jetson Nano, equipped with 128 CUDA cores and 4GB of LPDDR4 memory, delivers a peak computing power of approximately 472 GFLOPS. By integrating TensorRT's mixed-precision inference and operator fusion mechanisms, the system achieves a balance between computational efficiency and energy consumption in embedded environments. After quantization and pruning optimization, the model file is approximately 5 MB, the GPU memory usage is less than 2.5 GB, the theoretical inference latency ranges from approximately 30 to 40 ms, and a real-time detection rate of 25–30 FPS can be stably achieved. These performance metrics meet the application requirements of traffic signal detection in low-latency and continuous-response scenarios, demonstrating the system's good computing power matching and stability on the Jetson Nano platform. This section further quantifies the feasibility of the aforementioned deployment scheme on the Jetson Nano platform in terms of resource consumption and real-time performance metrics.

5. Conclusion

This study addresses the challenges of reduced traffic signal detection accuracy and the difficulty of real-time model inference on embedded devices under adverse weather conditions and nighttime scenarios. A lightweight embedded detection system based on YOLOX-Nano is designed. Via the collaborative optimization of structural pruning and mixed-precision quantization, the system reduces the number of model parameters by approximately 40% and computational complexity by approximately 35%, achieving low-power, low-latency real-time detection capabilities on the Jetson Nano platform.

At the algorithm level, this paper introduces depthwise separable convolutions and a lightweight path aggregation structure (PAN) into the YOLOX-Nano backbone network and employs a decoupled detection head to separate classification and regression tasks, thereby effectively enhancing the recognition performance for small targets while preserving high detection accuracy. At the data level, an augmented dataset encompassing meteorological scenarios such as raindrops, fog, low light, and strong reflections is developed to improve the model's adaptability to lighting variations and image degradation. At the deployment level, the PyTorch→ONNX→TensorRT conversion process is employed to achieve cross-framework migration and operator fusion-based acceleration of the model. The inference latency is approximately 30–40 ms, and stable real-time detection results of around 25 FPS can be output, verifying the deployability and engineering application value of the lightweight model in low-power devices [10].

In summary, based on theoretical analysis and design verification results, this paper establishes a complete technical path spanning model design, training optimization, to embedded deployment, providing an engineering solution for intelligent traffic signal detection under complex weather conditions. The system ensures detection accuracy while balancing power consumption and real-time performance, and has high practical value and promotion significance. Future work can be carried out in two aspects: first, further expand the dataset with samples of extreme weather conditions and nighttime intersections to improve the generalization robustness of the model; second, combine automated architecture search (NAS) and multi-sensor fusion strategies to optimize the deployment performance of the system on ultra-low-power platforms, providing a more efficient and reliable perceptual basis for intelligent transportation and vehicle-road cooperative applications.

References

[1] Wong A, Famuori M, Shafiee M J, et al. YOLO Nano: a Highly Compact You Only Look Once Convolutional Neural Network for Object Detection[J]. arXiv preprint arXiv:1910.01271, 2019.

[2] Ge Zhuangzhuang. Research on Traffic Lights and Digital Detection and Recognition Based on Embedded GPU [D].

ISSN 2959-6157

University of Electronic Science and Technology of China, 2020. DOI:10.27005/d.cnki.gdzku.2020.003623.Jayasinghe O, Hemachandra S, Anhettigama D, et al. Towards Real-time Traffic Sign and Traffic Light Detection on Embedded Systems[C]. IEEE International Conference on Industrial and Information Systems (ICIIS), 2022.

[3] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 779–788.

[4] Sandler M, Howard A, Zhu M, et al. MobileNetV2: Inverted Residuals and Linear Bottlenecks[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 4510–4520.

[5] Zhou W, Min X, Hu R, et al. FasterX: Real-Time Object Detection Based on Edge GPUs for UAV Applications[J]. arXiv preprint arXiv:2209.03157, 2022.

[6] Han S, Mao H, Dally W J. Deep Compression: Compressing Deep Neural Networks With Pruning, Trained Quantization and Huffman Coding[C]. International Conference on Learning Representations, 2016.

[7] Cui Lei, Chen Yiping. Real-time detection of pedestrian crossings based on improved YOLOX-Nano model in autonomous driving scenarios [J]. Journal of Guiyang University (Natural Science Edition), 2024, 19(4): 74-82.Jocher G, Chaurasia A, Qiu J. YOLOv5: An Improved Version of YOLO for Object Detection[EB/OL]. Ultralytics, 2020. https://github.com/ultralytics/yolov5

[8] Yang Yongbo, Li Dong, Fang Jiandong, et al. Lightweight traffic light detection algorithm for embedded terminals [J]. Computer Engineering and Applications, 2024, 60(13): 361-368.