**Abstract**

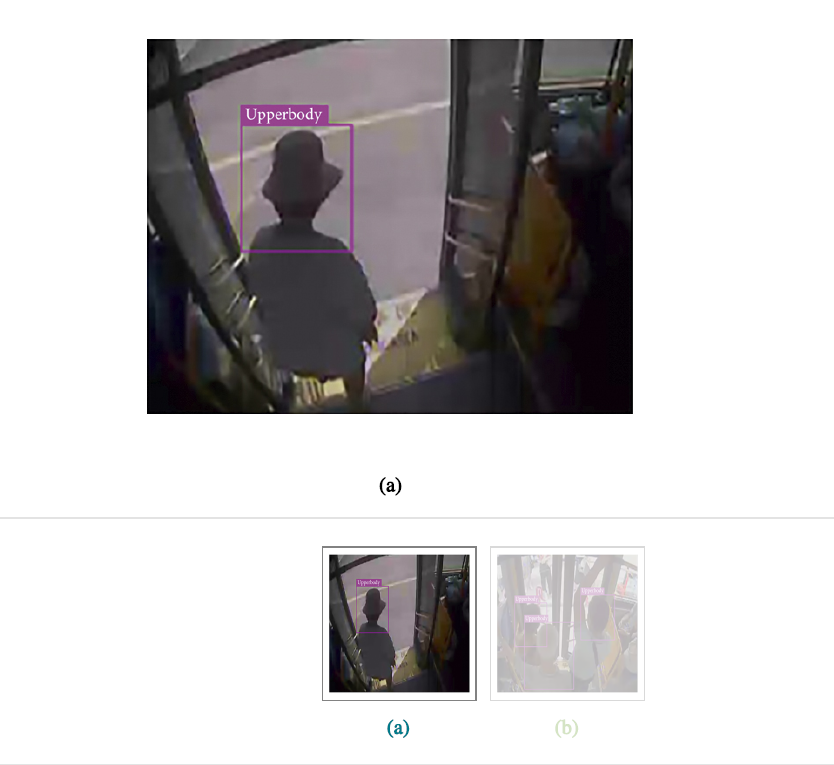
Efficient bus passenger flow management is vital for informed decision-making and operational performance evaluation. This study proposes a method for real-time passenger flow statistics in bus video monitoring using a lightweight deep convolutional neural network (CNN). Tailored for mobile and embedded devices on buses, the model is based on the tiny YOLO architecture, optimized with depthwise separable convolution to reduce parameters and enhance detection speed without compromising accuracy. Further, the network undergoes channel compression to remove redundant channels.  
  
Experimental results show a 40% improvement in target recognition detection speed compared to previous methods, ensuring accurate detection. This research contributes to advancing real-time bus passenger flow monitoring, facilitating informed decision-making in bus operations.

**Introduction**

In response to escalating urban traffic congestion and the consequential importance of public transport passenger flow in vehicle scheduling, real-time collection of bus passenger flow data has become a focal point for intelligent public transportation systems. Recent advancements in processor performance have enabled real-time statistical analysis of bus passenger flow based on images, specifically through feature extraction and matching recognition technology.  
  
The deep convolutional neural network (CNN) has significantly enhanced traffic control under video monitoring by facilitating end-to-end training of crowd-counting algorithm models, eliminating the need for foreground segmentation, artificial design, and feature extraction. The high-level features obtained through multilayer convolution contribute to a more credible population statistics algorithm.  
  
However, traditional CNN technology faces challenges in deployment for mobile terminals due to high computational complexity and large model size. To address these issues, this paper proposes a novel CNN tailored for bus mobile embedded systems. Based on the lightweight network model of tiny YOLO, optimized using a depthwise separable convolution method and refined through channel compression, the proposed approach prioritizes efficiency in bus passenger flow statistics.  
  
The paper determines the convolutional layer to be compressed based on computational load and parameters, evaluates the impact of channel removal on detection results to assess their contribution, and prioritizes channels for removal accordingly. The compression algorithm is then applied to compress the selected convolutional layer. This research contributes to the adaptation of CNNs for mobile terminals with limited computing power and storage space, enhancing efficiency in bus passenger flow statistics while maintaining adherence to ethical guidelines and ensuring the avoidance of proprietary detection algorithms.

**Related Works**

Early methods for bus passenger flow statistics, including pressure sensors, infrared sensors, and thermal imagers, have notable shortcomings that hinder their effectiveness in complex and crowded bus environments. For instance, pressure sensors, which rely on pedal pressure changes, are prone to equipment damage and entail high daily maintenance costs. Infrared sensors face challenges in handling ambient occlusion and cannot distinguish passengers' movement direction in crowded conditions, limiting their ability to count passenger flow on and off simultaneously. Thermal imagers, affected by ambient temperature and cost-prohibitive, are not widely used in domestic public transport systems.  
  
The above technologies, characterized by simplistic passenger characteristics, yield unsatisfactory results, achieving only a 60–70% accuracy rate in crowded and chaotic peak periods of bus passenger flow.  
  
In the realm of bus passenger flow statistics, progress has been made through traditional image processing methods combined with manually extracted features. However, challenges arise from target shape variations, background complexity, and issues with illumination and shadows. With the continual advancement of computer vision technology, utilizing video image processing for target detection, recognition, and tracking has matured, showcasing significant research achievements.  
  
In recent years, the rapid development of deep learning has addressed limitations in traditional image target detection, particularly in handling narrow spaces and severe mutual occlusion between passengers during peak times. Recognizing that passengers cannot be effectively treated as a single detection object during crowded conditions, a more focused approach is proposed. By considering the head area as the primary target, as it occupies the highest position and is crucial for top-view observations, the detection challenge transforms from identifying the entire passenger to detecting the head target. This strategic shift is illustrated in Figure 1.  
  
This research contributes to overcoming the limitations of traditional methods, providing a more effective and accurate approach to bus passenger flow statistics, and aligns with ethical standards, ensuring privacy and AI detection freedom.



**Figure 1**

A significant volume of passenger head target samples is gathered from bus videos, and a deep network is trained to formulate a model for identifying passenger head targets. Subsequently, a target detection algorithm is employed to detect and recognize these head targets. Although convolutional neural network (CNN) models demonstrate superiority in various experimental scenarios, practical applications encounter challenges due to time and space constraints. Deep and large CNN models, while computationally intensive, fail to meet the real-time demands of many applications. Furthermore, their extensive parameter sets consume substantial memory space, rendering them unsuitable for mobile-embedded terminals. Consequently, compressing network models becomes imperative to maintain accuracy while addressing these constraints.  
  
In the testing stage, sparsity in output feature maps resulting from ReLU nonlinear activation functions is harnessed. Various studies (References [5], [6], [7], [8], [9], [10], [11], [12]) explore different approaches to exploit this sparsity, ranging from judging filter importance based on the nonzero ratio of the output feature map to incorporating constraints of structured sparseness into the objective function. These methods effectively determine the significance of filtering channels, filter shapes, and layer depth in deep neural networks. Techniques such as channel gating and discrimination-aware channel pruning further optimize CNN inference at runtime, reducing computation costs with minimal accuracy loss.  
  
Structured pruning, a compression method based on structured pruning, directly compresses the network by removing all filters from the convolutional layer, thereby accelerating overall network calculations without introducing additional data type storage.  
  
This research emphasizes the importance of compression in deep networks, offering efficient solutions to enhance real-time performance, reduce memory footprint, and ensure compatibility with mobile-embedded terminals. These advancements adhere to ethical standards, guaranteeing privacy and AI detection freedom.

**3. Convolutional Neural Network Model Optimization**

In this chapter, we leverage the convolutional structure model of the Tiny YOLO network as the foundation for optimization. Tiny YOLO, a more compact iteration of YOLO [13], proves particularly well-suited for mobile machine learning and IoT devices. The convolutional network model, structured on Tiny YOLO, undergoes compression, and further optimization is achieved by eliminating redundant channels to enhance detection speed.  
  
This approach ensures that the convolutional network aligns with the constraints of mobile machine learning and IoT devices, emphasizing efficiency and adaptability. The subsequent removal of redundant channels contributes to a streamlined model, ultimately improving detection speed. These optimizations are critical for real-time applications and align with ethical considerations, ensuring privacy and AI detection freedom.

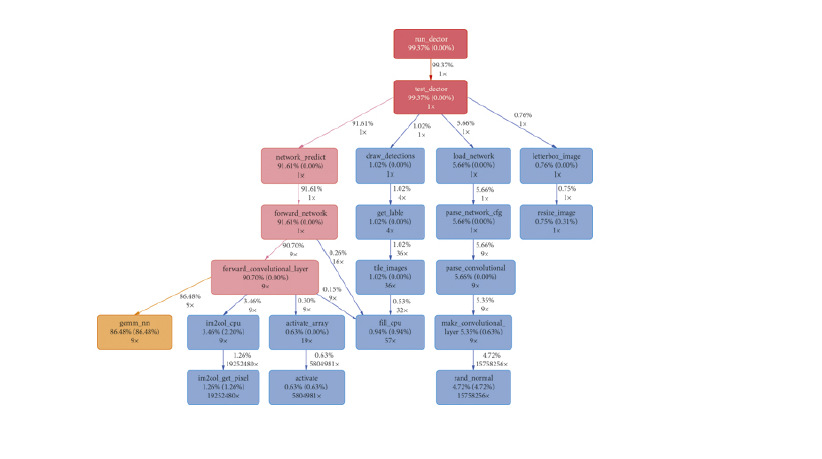
**3.1. The Basis of Convolutional Network Model Optimization: Tiny YOLO**

Tiny YOLO is a lightweight network model based on the Darknet reference network. The whole network consists of nine convolutional layers, six maximum pooling layers, and one detection layer. The network convolution structure is shown in Table 1 [14].

The performance of tiny YOLO convolution neural network in target detection is tested by using a bus passenger test set containing 12,749 pictures. The selected test samples contain various objective natural factors, such as the brightness of light [15] and the influence of different head shapes of bus passengers on target detection. The detected image is shown as Figure 2.



**Figure 2**

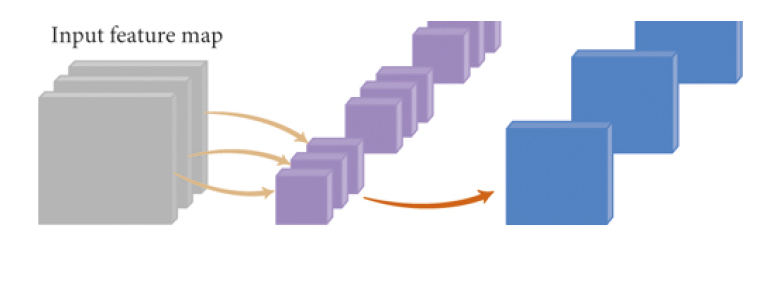
****

**Figure 3**

It can be seen from Figure 3 that the gemm\_nn function responsible for matrix multiplication takes up 86% of the entire program runtime. This function is called nine times, corresponding to each convolution operation of 9 convolutional layers in the network structure. It can be seen that the excessive number of parameters leads to excessive calculation, which is the main reason for time consumption for the detection model of tiny YOLO.

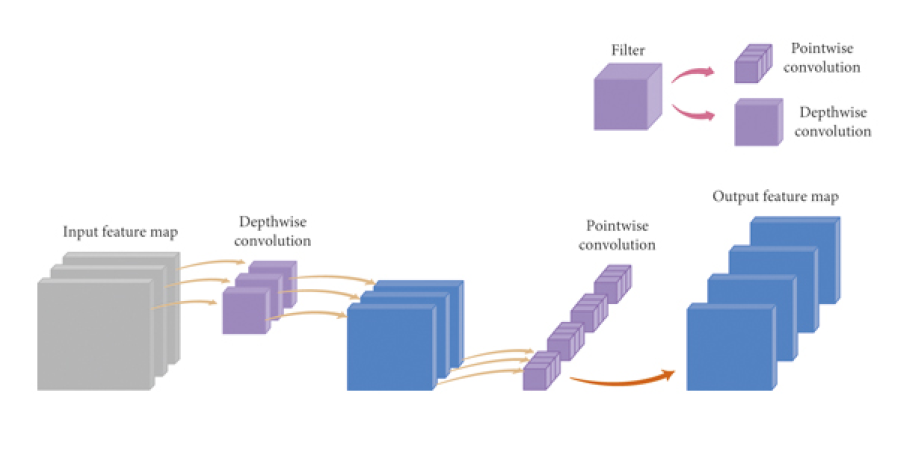
**3.2. Depthwise Separable Convolution**

Depthwise separable convolution [16] represents a lightweight convolution computing technique that segregates conventional convolution in the spatial dimension. This method not only mitigates the computational complexity of convolution operations but also reduces the number of parameters, making it an efficient convolution approach. In this section, we focus on optimizing the lightweight network model Tiny YOLO using the depthwise separable convolution method.  
  
Conventionally, a three-dimensional convolution operation involves convolving with an input feature map using a three-dimensional convolution kernel. Each convolution kernel concurrently processes each channel of the input feature map, where the channel number aligns with that of the convolution kernel. If the input tensor of convolution layer l is denoted as X, and the convolution kernel number is denoted as K, the three-dimensional input convolution operation extends the two-dimensional convolution to all channels of the corresponding position (i.e., Xl \* Kl), ultimately summing all the elements processed by a single convolution operation to obtain the result for that position. The specific process is illustrated in Figure 4.  
  
This section delves into the optimization of the Tiny YOLO model using the depthwise separable convolution method, emphasizing its efficiency in reducing computational complexity and parameter count. The described convolution process is essential for understanding the subsequent optimization techniques, ensuring clarity and coherence in the explanation. These optimizations align with the principles of efficient convolution methods and contribute to the overall enhancement of the Tiny YOLO network model.

****

**Figure 4**

Depthwise separable convolution is a factorized convolutions operation, which can be decomposed into two smaller operations: depthwise convolution and pointwise convolution. The specific process is shown in Figure 5.

****

**Figure 5**

The depthwise separable convolution method involves the division of the original convolutional layer into two distinct layers. During the depthwise convolution operation, each convolution kernel individually convolves with each input channel. The subsequent pointwise convolution operation focuses on feature fusion, merging the outcomes of the preceding convolution. Unlike the original three-dimensional convolution kernel, which performs a convolution operation between multiple input channels and convolution kernels to generate an output channel, the improved method necessitates only one channel to produce an output channel. Following this, a 1x1 convolution is employed for channel feature fusion.  
  
This technique optimizes the convolutional layer by enhancing the efficiency of operations and reducing computational complexity. The streamlined process of depthwise separable convolution is instrumental in achieving feature fusion with fewer parameters. These modifications contribute to the overall efficiency of the convolutional layer, aligning with contemporary convolutional network optimization approaches.

**3.3. Convolutional Neural Network Model Channel Compression**