electronics

MDPI

Article

One-Stage Small Object Detection Using Super-Resolved Feature Map for Edge Devices

Xuan Nghia Huynh, Gu Beom Jung and Jae Kyu Suhr *

Department of Intelligent Mechatronics Engineering, Sejong University, Seoul 05006, Republic of Korea; xnhuynh@sju.ac.kr (X.N.H.); gubeom620@gmail.com (G.B.J.)
* Correspondence: jksuhr@sejong.ac.kr; Tel.: +82-2-3408-3481

Abstract: Despite the achievements of deep neural-network-based object detection, detecting small objects in low-resolution images remains a challenging task due to limited information. A possible solution to alleviate the issue involves integrating super-resolution (SR) techniques into object detectors, particularly enhancing feature maps for small-sized objects. This paper explores the impact of high-resolution super-resolved feature maps generated by SR techniques, especially for a one-stage detector that demonstrates a good compromise between detection accuracy and computational efficiency. Firstly, this paper suggests the integration of an SR module named feature texture transfer (FTT) into the one-stage detector, YOLOv4. Feature maps from the backbone and the neck of vanilla YOLOv4 are combined to build a super-resolved feature map for small-sized object detection. Secondly, it proposes a novel SR module with more impressive performance and slightly lower computation demand than the FTT. The proposed SR module utilizes three input feature maps with different resolutions to generate a super-resolved feature map for small-sized object detection. Lastly, it introduces a simplified version of an SR module that maintains similar performance while using only half the computation of the FTT. This attentively simplified module can be effectively used for real-time embedded systems. Experimental results demonstrate that the proposed approach substantially enhances the detection performance of small-sized objects on two benchmark datasets, including a self-built surveillance dataset and the VisDrone2019 dataset. In addition, this paper employs the proposed approach on an embedded system with a Qualcomm QCS610 and demonstrates its feasibility for real-time operation on edge devices.

Keywords: one-stage detector; super-resolution; small object detection; embedded system; edge device

Citation: Huynh, X.N.; Jung, G.B.; Suhr, J.K. One-Stage Small Object Detection Using Super-Resolved Feature Map for Edge Devices. *Electronics* **2024**, *13*, 409. https://doi.org/10.3390/electronics13020409

Academic Editors: Alessandro Sebastian Podda and Livio Pompianu

Received: 16 November 2023
Revised: 10 January 2024
Accepted: 16 January 2024
Published: 18 January 2024

1. Introduction

Object detection constitutes a foundational task within the realm of computer vision, comprising two key steps: (1) identification of potential object locations, and (2) categorization of identified objects into distinct classes. Prior to the emergence of deep learning techniques, object detection relied on manually constructed methods for handcrafted feature extraction, drawing inspiration from human-centric object recognition [1].

In recent times, this domain has been advanced due to the remarkable evolution of deep learning algorithms. Because of those enhancements of deep-learning-based methods, the performance of object detection algorithms has greatly improved with two dominant approaches: the two-stage approach [2–4] with superior detection accuracy, and the one-stage approach [5–10] with an advantage of processing speed. Despite this, the precise detection of small objects within practical low-resolution images remains a challenging problem due to a low number of pixels, indistinguishable features, complicated background, limited context information, and occurrences of occlusion and truncation [11–13]. Among many techniques in previous surveys [11–13] to address these difficulties, super-resolution (SR) is one of the most representative solutions. In particular, super-resolving an intermediate

feature map gains more efficiency than super-resolving an image directly. However, this approach has been applied to only two-stage detectors, which causes difficulties for use in real-time embedded systems. Therefore, integrating the intermediate feature-map-based SR technique into one-stage detectors becomes significant for small object detection in wide-ranging real-time applications such as autonomous driving, visual surveillance, remote sensing, etc.

Put simply, SR methods are designed to restore high-resolution features from corresponding low-resolution features, thereby augmenting the finer details of the original scene and intermediate features. The refined features contain richer information, making them well-suited for precisely detecting small objects. In initial practices, refs. [14–17] integrated a preceding SR sub-network with a detection sub-network to directly super-resolve the input image and put it into detectors. Subsequently, thanks to the development of a generative adversarial network (GAN) [18–21], GAN variants have been utilized to generate SR images, which yields improved performance in small object detection. The authors of [22,23] provided more tailored strategies emerging in the form of a two-stage approach. In those, GANs were employed selectively to super-resolve only regions potentially containing small objects, enhancing their detectability. However, while input image super-resolution techniques offer advantages and can be easy to apply in any manner, they are also accompanied by drawbacks: (1) the need for two separate networks for different tasks, incurring expensive computation costs; (2) slower processing speeds, limiting practical application; and (3) the utilization of a large-sized parameter-heavy model, restricting the use of resource-constrained embedded systems.

In the other works, the concentration shifted to super-resolving intermediate feature maps to mitigate the shortcomings of image-based SR. However, this has only been applied to two-stage approaches. This approach aims at super-resolving the features of small objects and generating refined features that conductively produce accurate predictions of small objects. The authors of [24,25] initially generate proposals in the first stage. In the second stage, the features of those regions potentially containing small objects are super-resolved using GAN-based training strategies, ultimately producing final predictions. Diverging from GAN-centric methods, ref. [26] introduced a novel SR module named Feature Texture Transfer (FTT) that generates a super-resolved feature map tailored to the small-scale detection head. While these methods exhibited the potential for high detection rates and precise localization of small objects, they also inherited some of the disadvantages associated with two-stage approaches: (1) sophisticated architecture with many stages; (2) complex training and inference procedures; and (3) elevated computational demands as the number of proposals increased. These limitations have restrained the applicability of two-stage methods in real-world scenarios, particularly in embedded systems characterized by limited computational resources and the need for instant response. On the contrary, the one-stage approach demonstrates noticeable strides, characterized by simpler network architectures, straightforward training processes, and rapid processing speeds. Leveraging feature-based SR in conjunction with a one-stage detector holds significant promise for deployment in resource-constrained real-time embedded systems and effectively tackles the challenges of small object detection.

This paper proposes a novel method that efficiently leverages feature-based SR within a one-stage detector. Firstly, it adopts the FTT module [26] originally designed for use in the two-stage detector [10] in the framework of the one-stage detector. This SR module functions as a fusion mechanism combining two feature maps: the main feature with rich semantic insight and the reference feature with shallow contextual information. The fusion results in a refined feature map that enhances the accuracy of detecting small objects and lowers the computational burden associated with direct high-resolution input image usage. Secondly, this paper proposes a novel SR module that extends the input into three feature maps. Compared with the FTT module, the proposed one utilizes a shallower feature map derived from the backbone with one more integration to synthesize the super-resolved feature map. Via the dual integration of three inputs, it compresses more detailed contextual

information of small objects, surpassing the efficiency while maintaining the approximate computational burden of the FTT module. The proposed approach that uses the SR module with the one-stage detector preserves the outstanding properties of the one-stage detector to save more memory and facilitates end-to-end training. Lastly, this paper suggests a simplified version of the proposed SR module to enhance the excellent aspects above for real-world detection applications. The integration of this simplified module delivers similar performance while efficiently halving computational cost compared to the FTT module. In experiments, the proposed approach was evaluated by both public and self-built datasets and improved object detection performance, especially for small-sized objects. In addition, this approach was successfully embedded into an edge device equipped with Qualcomm's neural processing unit (NPU) to show its real-time operability.

The main contributions of this paper can be summarized as follows:

- It introduces the integration of the FTT, originally designed for use in two-stage detectors, into a one-stage detector. This integration serves to improve the detection performance of small objects.
- It proposes an SR module that leverages three distinct input feature maps and synthesizes information twice, generating a super-resolved feature map tailored to the small-sized specific detection head. This approach enhances performance while upholding computational efficiency compared to the FTT module.
- It suggests a simplified version of the SR module that achieves similar performance as the FTT module while concurrently halving computing resources, making it similar to the vanilla one-stage detector.
- It shows that the proposed approach can be efficiently embedded into an edge device with an NPU for real-time processing.

2. Related Works

Deep learning techniques have emerged as powerful tools for general object detection, driven by their remarkable performance. Two-stage methods [2–4] generate Regions of Interest (RoIs) followed by classification and precise localization. Meanwhile, one-stage methods [5–10] directly perform classification and localization simultaneously, often employing pre-defined anchor boxes. Despite the development of deep-learning-based object detectors, the detection of small objects remains a challenging task. Successful small object detection has four key aspects: multi-scale representation, contextual information, region proposal, and super-resolution [11].

In the case of multi-scale representation, repeating down-sampling operations and pooling layers results in the loss of small object information and produces the final feature map with a large receptive field and strong semantic information but in a low resolution. This contributes to poor detection of small objects. To address this problem, ref. [27] used deconvolution to fuse different-scale feature maps and generate a higher-resolution feature map for detection heads. In [28], the Single-Shot Detector (SSD) extends up to seven heads at different scales via a fusion block to detect more small objects. The authors of [29] combined Faster RCNN [3] and Feature Pyramid Network (FPN) [10], and [4] combined ResNet with FPN via lateral connection to produce multiple scale-specific feature maps for detection. Inspired by FPN with a bottom-up path aggregation, ref. [30] proposed a Path Aggregation Network (PANet), which additionally supplements a top-down path aggregation to incorporate twice and generate pyramid scale-specific feature maps. These works fuse multi-scale feature maps and produce high-resolution feature maps with more detailed information, facilitating small object location and classification.

In the case of contextual information, small objects only occupy a relatively small portion of the image, constraining the extraction of meaningful information. Adaptive convolution [31] and dilated convolution [32] are leveraged to capture interactions between objects and their surroundings, boosting detection capabilities, particularly for small objects. In the case of the region proposal, large anchor sizes in Faster R-CNN [3] and R-FCN [33] reduce the detection performance of small objects. To mitigate the issue, [34] generated an

additional higher-resolution feature map for small objects with smaller anchor sizes, then reduced the number of RoIs by adopting a scale-specific objectness attention mechanism. The authors of [35] applied an area proposal network to crop the regions that contain at least one object and enlarge those regions to make small objects easier to detect.

In the case of super-resolution, the methods aim to recover high-resolution features from corresponding low-resolution features, offering better conditions to detect small objects. Our proposed method is categorized into this domain; therefore, we have conducted a deeper literature review of this approach.

Super-resolution techniques for small object detection can be categorized into two primary branches: image-based super-resolution and feature-based super-resolution. The former involves super-resolving an image into a higher-resolution version, which enlarges the object scale for better small object detection. In early practices, high-resolution images in [36] are obtained by applying bilinear interpolation to enlarge and detect small human faces. Then, ref. [14] shared the same idea with JCS-Net [15], which incorporated an SR-subnet for direct input super-resolution with a primary task subnet for classification or detection in a unified framework. Moreover, ref. [17] adopted the same methodology and added an additional feature-based loss derived from knowledge distillation technology. In another way, ref. [16] used an SR-module FTT [26] instead of a dedicated SR-subnet to directly super-resolve input and extract RoIs of small faces from that in the first stage. However, the recovered features tend to be blurred and not photorealistic when applying those methods. With the advent of Generative Adversarial Networks (GANs) [18], they have become a representative method for generating super-resolution images. For vehicle detection in remote sensing images, ref. [19] designed a joint network of a sub-network similar to MsGAN (multi-scale GAN) for generating super-resolution images, and the YOLOv3 detector for object detection. In particular, super-resolution GAN (SRGAN) [37] introduced GAN into the super-resolution image generation task, only relying on a perceptual loss. Then, ref. [21] utilized SRGAN to upscale images with more distinguishable features between background and pedestrians, before applying Faster R-CNN [3] to improve small pedestrian detection. More generally, ref. [20] changed MsGAN into super-resolution Wasserstein GAN (SR-WGAN) designed for SR tasks to detect small objects in remote sensing images. Nevertheless, image-based super-resolution encounters the critical issue of redundant information generation because it super-resolves both the foreground and useless background in the image. For the two-stage approach, ref. [23] used a GAN-based network to super-resolve RoIs of small faces from the RPN and classify them as face or non-face patches. Following the same idea for common small objects, ref. [22] proposed a multi-task GAN (SOD-MTGAN), where the discriminator served as a multi-task network for real/fake authentication of RoIs, classification, and regression from RoIs. Still, there are associated drawbacks, such as heavy and complicated architecture, the need for paired images, and the burden of computation and memory when the number of RoIs increases.

For feature-based super-resolution, intermediate feature maps are super-resolved to enrich small object features and improve detection performance, but this approach has only been applied to two-stage detectors. One of the pioneering techniques is Perceptual GAN [24], which generates super-resolved features of proposals related to small traffic signs to attenuate the differences from large ones. Additionally, ref. [25] adopted the supervision technique into a similar GAN-based strategy to enhance the process. Differently, ref. [26] introduces the FTT module to generate the entire new feature map of a specific small-scaled detection head in the two-stage FPN detector [10]. This method improves geometric details and context information via super-resolution and distillation techniques. Despite the improvements in small object detection, they inherit the disadvantages associated with two-stage detectors, including slower response times and higher memory consumption. Therefore, combining one-stage detectors and feature-based super-resolution techniques emerges as a solution. It is a prospective domain as one-stage detectors are known for their advantages in terms of speed, computational efficiency, and memory usage, while the feature-based super-resolution excels at direct feature reconstruction for small objects.

The previous and proposed methods can be summarized by the hierarchical diagram shown in Figure 1. As aforementioned, small object detection methods are mainly categorized into four approaches: multi-scale representation-based, contextual information-based, region proposal-based, and super-resolution (SR)-based techniques. The SR-based approach consists of image-based SR methods and feature-based SR methods. While the feature-based SR methods have been implemented based on the two-stage detectors, this paper proposes a combination of the one-stage detector and the feature-based SR. Inherited from the advantages of the one-stage detector, this combination is advantageous for use in real-time embedded systems.

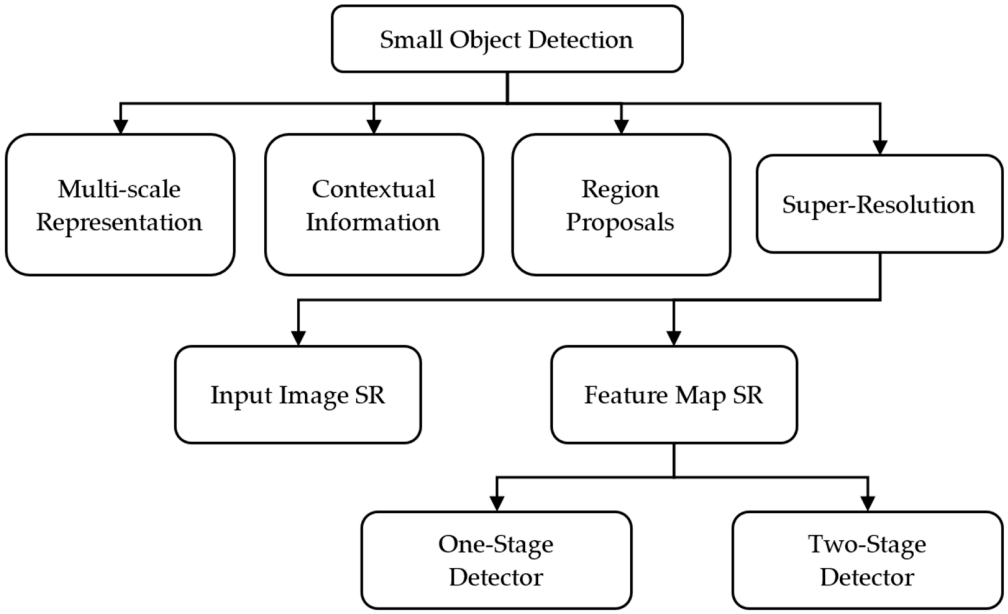

Figure 1. Hierarchical diagram of the related works: Small Object Detection [11]; Multi-scale Representation [4,27–30]; Contextual Information [31,32]; Region Proposals [34,35]; Input Image SR [14–17,19–23]; One-stage Detector (Proposed Method); Two-stage Detector [24–26].

3. Proposed Method

The proposed method is built along with a careful analysis of the impact of low-level high-resolution feature maps in small object detection. As shown in [38], shallow low-level features responsible for detecting small objects tend to be less discriminative due to excessive background noise. Conversely, high-resolution feature maps, as highlighted in [39], play a crucial role in increasing the localization accuracy of small objects. Our investigation revolves around understanding how a shallow high-resolution feature map can compensate for the information loss of small objects. More specifically, this paper focuses on enhancing the feature map responsible for detecting small objects by combining a one-stage detector with SR modules, which generate super-resolved feature maps. In Section 4.1, we suggest how the FTT module can be inserted into the one-stage detector as an SR module. In Sections 4.2 and 4.3, we propose a novel SR module and its simplified version, respectively. As a base one-stage detector, this paper utilizes YOLOv4 [8]. This detector was chosen because it has been proven useful in various applications for a considerable time by demonstrating a compromise between detection accuracy and computational efficiency in diverse frameworks [40,41]. In addition, it has been successfully embedded into an NPU and shown to perform in real-time [42].

3.1. FTT Module in One-Stage Detector

Prior to the use of the FTT module, this paper first builds a high-resolution (HR) variant of the vanilla YOLOv4 (YOLOv4-HR) by adding one more upper layer of the feature map in the neck and relocating the detection heads. This can be considered as the

simplest way to enhance the small object detection performance. In vanilla YOLOv4 (left in Figure 2), the detection heads (D3, D4, and D5) are connected to M3, N4, and N5, but in YOLOv4-HR (right in Figure 2), the detection heads (D2, D3, and D4) are connected to M2, N3, and N4 whose resolutions are twice that of M3, N4, and N5, respectively. In YOLOv4-HR, M2 and D2 are the feature map and the detection head responsible for small objects, respectively. To enhance the small object detection performance, we suggest a way to super-resolve the feature map responsible for detecting small objects by adopting the FTT module, which was originally developed for a two-stage object detector. It serves a dual purpose: simultaneously super-resolving low-resolution features and extracting regional textures from high-resolution features. This combination of operations enhances the feature map, making it better suited for detecting small objects.

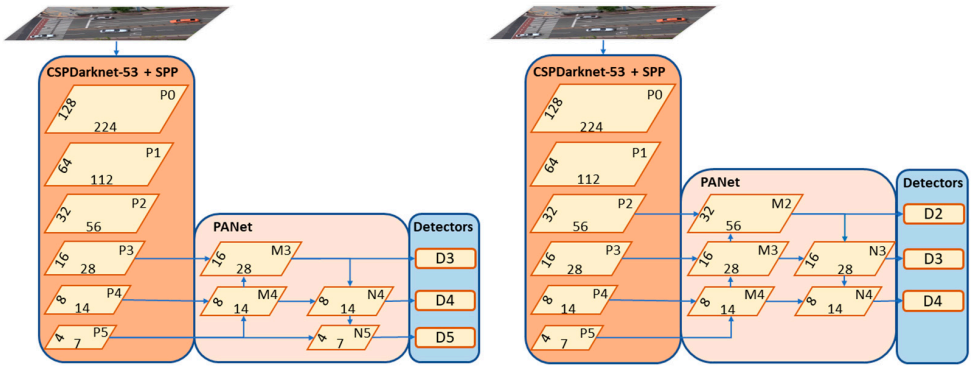

Figure 2. Architectures of YOLOv4 (**left**) and YOLOv4-HR (**right**).

Figure 3 shows a combination of YOLOv4-HR and the FTT module. In the left side of this figure, the FTT module super-resolves the feature map M3 from the neck based on the feature map P2 from the backbone, which contains critical texture information of small objects. The right side of Figure 3 shows the FTT module in detail. In this figure, residual blocks are employed to M3 to capture strong semantic information and manipulate the number of channels as needed. The sub-pixel convolution with the pixel shuffling technique inside is operated on the generated feature map to upscale the spatial resolution considering its efficiency. The pixel shuffle operator rearranges pixels on the dimension of channel into the dimension of width and height, which super-resolves a low-resolution feature map $F \epsilon \mathbb{R}^{H \times W \times 4C}$ into a high-resolution feature map $F' \epsilon \mathbb{R}^{2H \times 2W \times C}$. The concatenation of the pixel shufflement and P2 is fed into residual blocks again to pick up credible texture information of small objects and discard disturbing noises. Under the element-wise addition, the FTT output synthesizes both information of the semantics and textures of small objects for better detection.

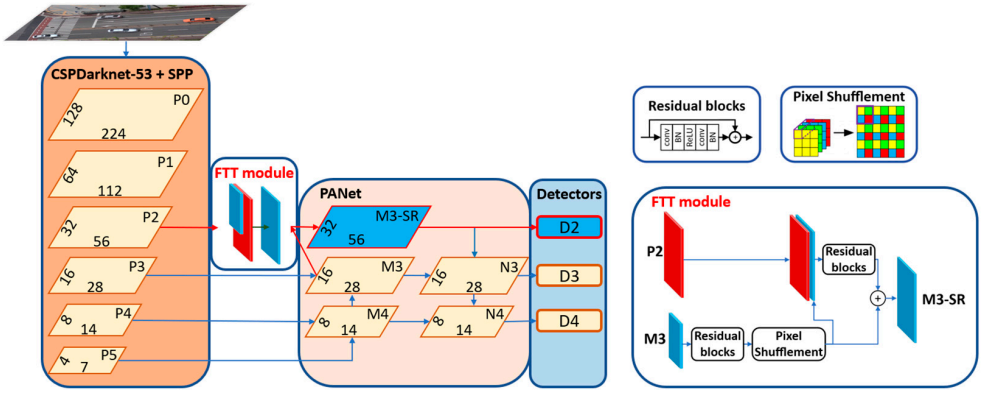

Figure 3. YOLOv4-HR with the FTT module (**left**) and detailed architecture of the FTT module (**right**).

3.2. Proposed SR Module (SRm)

The backbone CSPDarknet53 of YOLOv4 still has a higher-resolution feature map P1, which is not utilized to enhance performance. This feature map contains shallow information with a lot of noise but critical detailed information about small objects which is intensively filtered out during convolutions to high-level feature maps. With the help of the feature map P1, we can detect small objects in low-resolution images even better. Drawing inspiration from the FTT module as well as the super-resolution network SRGAN [37], which upscales the image twice and super-resolves the resolution four times larger, we propose an SR module that even makes use of feature map P1. This module is superior to the FTT module and powerfully enhances the feature map to obtain the capability of detecting small and tiny objects. The proposed SR module is illustrated in Figure 4.

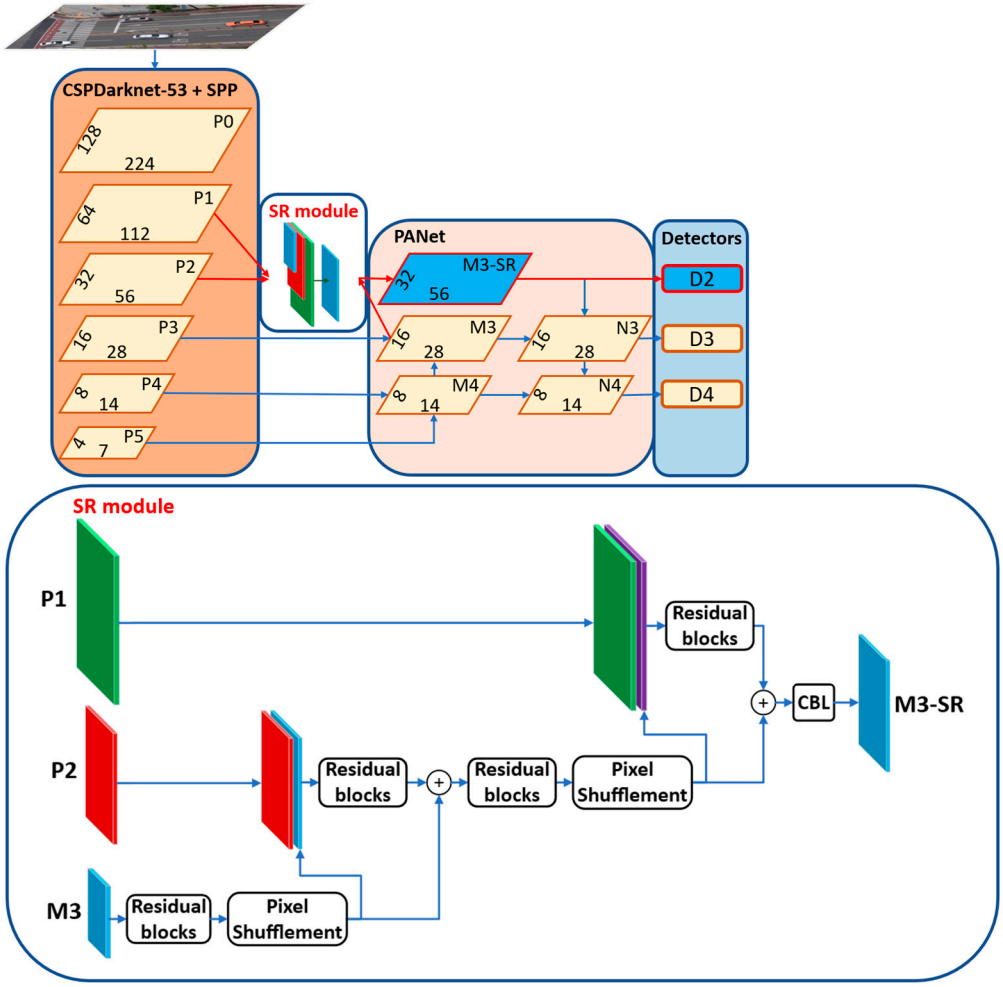

Figure 4. YOLOv4 with the proposed SR module (**top**) and detailed architecture of the proposed SR model (**bottom**).

At the top of this figure, the proposed SR module super-resolves the feature map M3 based on the feature maps P2 and P1 from the backbone, where P2 contains texture information with noise and P1 holds critical detailed information about small objects with more noise. The bottom of Figure 4 shows the proposed SR module in detail. For the first upgrade, M3 is fed into residual blocks to extract semantic information together with channel manipulation, followed by the pixel shuffle operation. An element-wise addition operation is performed after applying residual blocks to the concatenation of the super-resolved M3 and P2. For the second upgrade, the resulting feature map of the first upgrade is super-resolved again based on P1. The second upgrade uses the same strategy as the first upgrade. Because the resolution of the final super-resolved feature map

is double that which is needed and still contains noise, we apply a convolution with stride 2 (CBL in Figure 4) to decrease its resolution and filter out noises one more time to obtain the final best output feature map M3-SR. CBL indicates a combination of convolution, batch-normalization, and leaky ReLU. In summary, the output feature map aggregates three different input feature maps iteratively which contain rich semantic information, abundant contextual information, and credible detailed information simultaneously under the support of an adaptation layer to match the output feature map to the corresponding specific small-sized head.

3.3. Simplification of SR Module (SSRm)

To implement the proposed object detector into real-time embedded systems, we developed a simplified version of the SR module that benefits from computation cost and memory assumption with acceptable detection performance. The simplified SR module is the same as its original version, except that the residual blocks are replaced by a convolution before the super-resolving feature map by the pixel shuffle operator, shown in Figure 5. In addition, a convolution compression technique is leveraged within the residual blocks, which uses 3×3 convolution to reduce the number of channels and 1×1 convolution to restore again iteratively. Thanks to those techniques, the simplified module only consumes half of the computation costs compared to the FTT module while maintaining a similar detection performance, as later explained in the experimental section. Since the simplified SR module aggregates the information from three different feature maps with a slight cost increase, it can be used as a practical SR module for resource-constrained real-time embedded systems.

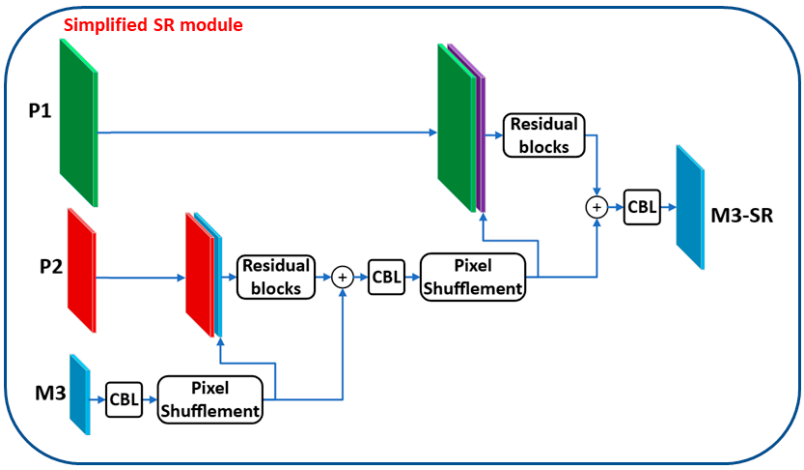

Figure 5. Detailed architecture of the simplified SR module.

4. Experiments

4.1. Datasets

The proposed small object detection method has two main target applications: drones and visual surveillance, both requiring edge computing. Thus, experiments were conducted with two datasets related to these applications: VisDrone2019 [43] and a self-built surveillance camera dataset. Table 1 shows the summary of the two datasets. In this table, instances are categorized as very tiny, tiny, and small following the AI-TOD dataset [44] criteria. The category of instances is defined by the number of pixels they occupy (very tiny: 2×2 to 8×8 pixels, tiny: 8×8 to 16×16 pixels, small: 16×16 to 32×32 pixels). Figure 6 shows example images and Table 1 summarizes the information of the two datasets.

Table 1. Summary of self-built and VisDrone2019 datasets.

Dataset		Self-Built	VisDrone2019
Number of images	Training set	21,494	6471
	Test set	3229	1610
Test set	Inference resolution (pixels)	224 × 128	480 × 288
	Total objects	15,794	75,102
	Very tiny objects	9086 (57.53%)	36,161 (48.15%)
	Tiny objects	4054 (25.67%)	20,321 (27.06%)
	Small objects	2209 (13.99%)	9386 (12.50%)

Figure 6. Example images of two datasets.

The self-built dataset includes surveillance camera images with three object classes: vehicle, pedestrian, and cyclist. It contains over 24,000 surveillance images with over 120,000 instances at various sizes. At the inference resolution of the test set, over 97% of instances cover an area of less than 32 × 32 pixels. The dominant majority of small objects makes it an appropriate benchmark for small object detection.

The VisDrone2019 dataset [43] is a widely used large-scale benchmark for small object detection. It consists of 8629 diverse-resolution images captured by drone platforms in different places at different heights. More than 540,000 instances are annotated with ten object classes: pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, and motor. Over 87% of objects in the dataset occupy less than 32 × 32 pixels at the inference resolution of the test set. The pedestrian and person classes pose the most difficult challenges since the instances are tiny and appear in crowds. These properties result in a proper benchmark for small object detection.

4.2. Evaluation Metrics

For comprehensive evaluation, this paper utilizes the mean Average Precision (AP) metric to measure detection performance, which is popularly used in object detection. The metric takes into account two distinct tasks in object detection: classification and localization. Average precision is calculated over all conference thresholds with class-based independence to remove the relevance to confidence. The mean of the average precisions of different classes, when IoU goes from 50% to 95% in 5% increments, is calculated and considered AP. This metric is based on both IoU and predicted classification scores, so it is fundamentally primary to measure general object detection performance. However, the IoU of a small object is unstable because a small change in IoU can result in a large

difference of AP. Thus, we also focus on AP at IoU = 0.50 (AP50) as a fundamental metric where it maintains primary evaluation while minimizing the effect of IoU.

Additionally, this paper collects information about the number of parameters in each model and the computation cost to illustrate the heaviness of the model. They are important, especially for embedded systems, to demonstrate the capability of the model in practical applications where resources are constrained. Last but not least, processing time and frames per second (FPS) are also utilized to evaluate the running speed of the algorithm on specific computation platforms.

4.3. Implementation Details

The input images are resized to 224 × 128 pixels and 480 × 288 pixels for the self-built and VisDrone2019 datasets, respectively. The backbone networks of all models are CSPDarknet53 initialized by the pre-trained weights on the MS COCO dataset. The anchors are generated using a K-means clustering algorithm from the training set. Three methods were used for data augmentation: random crop, random horizontal flip, and random translation. The losses are the same as those of the vanilla YOLOv4.

The networks were trained for 100 epochs with two warm-up epochs. The batch size is set to 16 for the self-built dataset and 6 for the VisDrone2019 dataset. Networks are optimized by the Adam optimizer, the β_1, β_2, and ϵ of which are set to 0.9, 0.999, and 10^{-7}, respectively. The learning rate initializes at 0, increases to 10^{-4} in warm-up epochs, and follows the cosine annealing scheduler to diminish continuously to 10^{-6} until the end of training. All the experiments are conducted using the TensorFlow framework. The training was performed on the desktop-based platform with Intel i7-12700K CPU and RTX 2080Ti GPU (Intel, Santa Clara, CA, USA).

In addition, the proposed method was deployed into an embedded system with Qualcomm QCS610 Systems-on-Chip (SoC), a high-performance SoC delivering premium features for building advanced smart cameras and Internet-of-Thing use cases encompassing machine learning as well as edge computing. This SoC integrates the CPU, GPU, and DSP for accelerated AI performance. In this paper, the models were implemented on the DSP of the QCS610 SoC. The process of embedding contains three main steps: (1) freeze the pre-trained model and convert it into a deep learning container (DLC) file with the data type as the floating-point32 (FP32); (2) quantize the model in the DLC file into fixed-point8 based on a post-training quantization approach; and (3) deploy into DSP chipset and run the model. This process is conducted by the Snapdragon Neural Processing Engine (SNPE) Software Development Kit (SDK) version 2.12. Figure 7 shows an AI camera that includes an embedded board with a Qualcomm QCS610 SoC. Because this camera can detect objects by itself without needing any other processing units, it can be effectively used as an edge device.

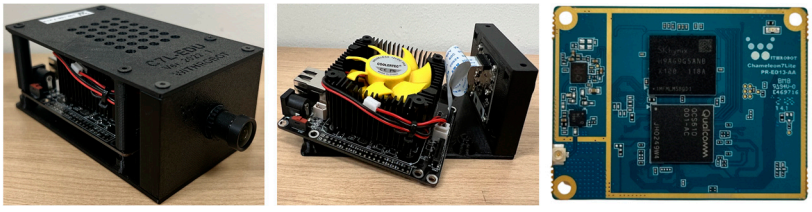

Figure 7. AI camera that includes an embedded board with Qualcomm QCS610 SoC; (**left**) exterior, (**middle**) interior, (**right**) embedded board inside the camera.

4.4. Results and Comparisons

This paper compares the performance of seven object detection networks as shown in Tables 2 and 3. In these tables, YOLOv8-L is the large version of YOLOv8 [45], one of the state-of-the-art detectors, which is developed and published for open usage by Ultralytics. SRGAN + YOLOv4 is a combination of the SRGAN and YOLOv4, where the SRGAN first super-resolves input images to double their sizes, and the vanilla YOLOv4 is applied

to these super-resolved input images. Because SRGAN + YOLOv4 uses two separate networks, its computation cost is much higher than the others. YOLOv4-HR is shown on the right side of Figure 2 and uses one more upper layer of the feature map for small object detection. YOLOv4-FTT is YOLOv4 with the FTT module in Figure 3. YOLOv4-SRm and YOLOv4-SSRm are YOLOv4 with the proposed SR module and its simplified version, as shown in Figures 4 and 5, respectively.

Table 2. Detection performance of seven networks on self-built dataset.

Model	Metrics							
	AP50	**AP**	**AP50$_{VT}$**	**AP$_{VT}$**	**AP50$_{T}$**	**AP$_{T}$**	**AP50$_{S}$**	**AP$_{S}$**
Vanilla YOLOv4	67.71	38.62	48.18	19.36	88.59	52.78	94.41	70.05
YOLOv8-L	67.34	41.84	45.98	19.71	88.75	54.16	96.88	76.16
SRGAN + YOLOv4	79.17	48.05	69.20	33.14	93.35	61.71	96.56	75.53
YOLOv4-HR	72.66	41.52	60.25	25.76	89.08	55.16	92.96	68.26
YOLOv4-FTT	77.31	44.15	66.51	28.43	92.37	57.28	93.78	71.37
YOLOv4-SRm	79.09	47.42	69.75	32.60	92.14	60.20	95.46	72.06
YOLOv4-SSRm	77.20	44.03	65.86	28.76	91.84	56.43	96.60	71.69

Table 3. Detection performance of seven networks on VisDrone2019 dataset.

Model	Metrics							
	AP50	**AP**	**AP50$_{VT}$**	**AP$_{VT}$**	**AP50$_{T}$**	**AP$_{T}$**	**AP50$_{S}$**	**AP$_{S}$**
Vanilla YOLOv4	15.98	7.23	3.14	0.95	17.77	6.56	34.46	15.15
YOLOv8-L	19.08	9.59	4.36	1.28	22.22	10.01	42.11	20.51
SRGAN + YOLOv4	26.79	13.28	8.97	3.04	32.88	14.68	48.48	25.56
YOLOv4-HR	21.02	9.66	5.86	1.77	25.20	9.80	40.52	19.38
YOLOv4-FTT	22.78	11.02	7.19	2.42	29.19	12.15	43.06	21.64
YOLOv4-SRm	24.86	12.28	8.87	3.04	30.65	13.46	43.53	23.15
YOLOv4-SSRm	22.27	10.98	7.60	2.48	27.31	11.57	39.66	21.13

Table 2 summarizes the detection performance on the self-built test set. It can be easily noticed that the three proposed networks (YOLOv4-FTT, YOLOv4-SRm, and YOLOv4-SSRm) outperform both vanilla YOLOv4 and YOLOv8-L by about 10~12% in terms of AP50. In the case of very tiny objects, the performance gaps even increase. In terms of AP50$_{VT}$ for very tiny objects, the performance gaps are about 17~21%. In terms of AP50$_{T}$ for tiny objects, the performance gaps are about 3~4%. Although YOLOv8-L shows better detection performance than vanilla YOLOv4, the three proposed networks still outperform it. Among the three proposed networks, YOLOv4-SRm shows the best performance, and YOLOv4-FTT and YOLOv4-SSRm show similar performance. Even though SRGAN + YOLOv4 performs slightly better than the three proposed networks, its computational cost is much higher than the others by about 5~10 times. Details of the computational cost will be discussed later in this paper. Compared to YOLOv4 with the FTT module (YOLOv4-FTT), YOLOv4 with the proposed SR module (YOLOv4-SRm) provides a higher AP50$_{VT}$ for very tiny objects by about 3%. This reveals that using one more high-resolution feature map in the case of the proposed SR module can help detect very tiny objects compared to the FTT module.

Table 3 shows the detection performance on the VisDrone2019 test set. The same performance tendency as in Table 2 can also be found in this table, even though the performance of all methods are lower than those of Table 2 because the VisDrone2019

dataset is much more challenging than the self-built dataset. The results in Tables 2 and 3 clearly show the SR modules play vital roles in one-stage detectors for finding small objects because they can generate high-resolution feature maps that focus more on gaining information about small objects. Figures 8 and 9 illustrate the example detection results of the proposed networks (YOLOv4-FTT, YOLOv4-SRm, and YOLOv4-SSRm) on the self-built and VisDrone2019 datasets, respectively. In these figures, it can be noticed that the proposed methods precisely detect small objects as well as objects of other sizes. In particular, they could correctly predict and distinguish between similar instances (pedestrian and cyclist in the self-built dataset and people and pedestrian in the VisDrone2019 dataset).

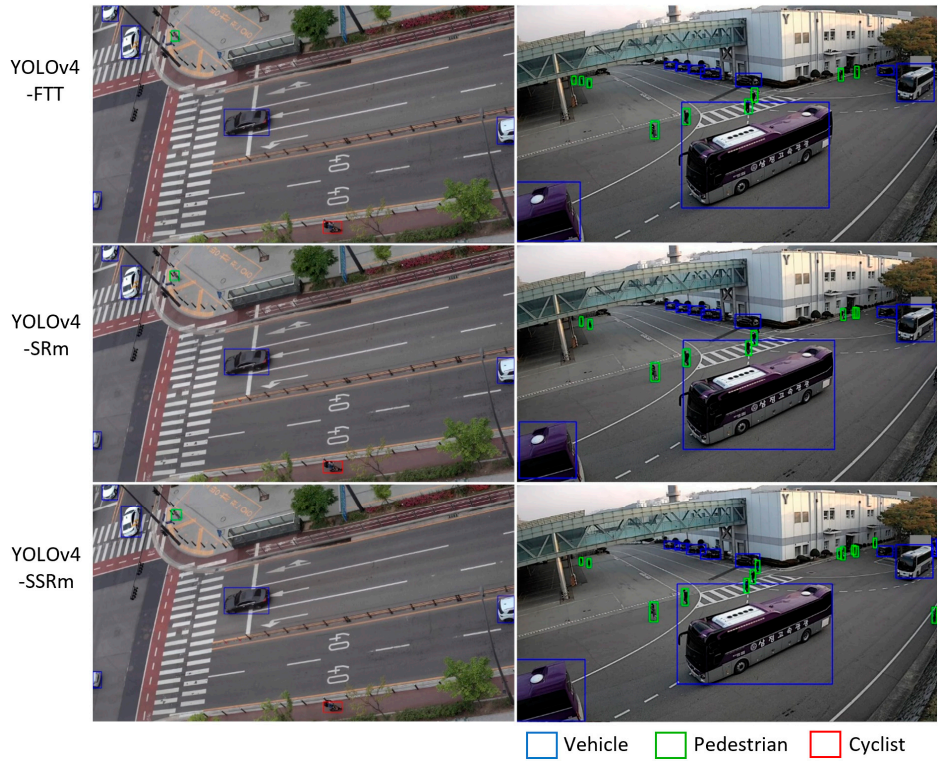

Figure 8. Detection results of the proposed methods on self-built dataset images.

Along with performance enhancement, the proposed networks achieve real-time processing capacities in both desktop and embedded environments. Table 4 shows model sizes, computational costs, and inference times of seven networks in the desktop environment. Regarding the giga floating point operations per second (GFLOPs), SRGAN + YOLOv4 requires about five times more operations than YOLOv4-FTT and YOLOv4-SRm. YOLOv4-SSRm and YOLOv8-L only need approximately half the operations compared to YOLOv4-FTT and YOLOv4-SRm. This means that compared to the vanilla YOLOv4 and YOLOv8-L, YOLOv4-SSRm requires almost the same computational cost but provides 10% higher AP50 and 17~20% higher $AP50_{VT}$ on the self-built dataset. Thus, YOLOv4-SSRm can be a good compromise between detection performance and computational cost from the viewpoint of real-time embedded systems. Despite the differences in GFLOPs, inference times of all methods except SRGAN + YOLOv4 are similar to each other in Table 4. This is because of the parallel processing ability of the high-end GPU attached to the desktop. However, the differences in GFLOPs clearly affected the inference times in the embedded environment, where the computational resources are restricted. Table 5 shows the inference times in the embedded system with the Qualcomm QCS610 SoC in Figure 7. In this table, SRGAN + YOLOv4 is excluded because the SRGAN cannot be implemented in this environment due to unsupported operations it uses, and there is no need to do so because it cannot operate in real time. Unlike the desktop environment, differences in the inference times can be clearly seen in Table 5. It shows that YOLOv4 with the proposed simplified

SR module (YOLOv4-SSRm) can handle more than 30 images per second, which definitely satisfies the condition of real-time processing.

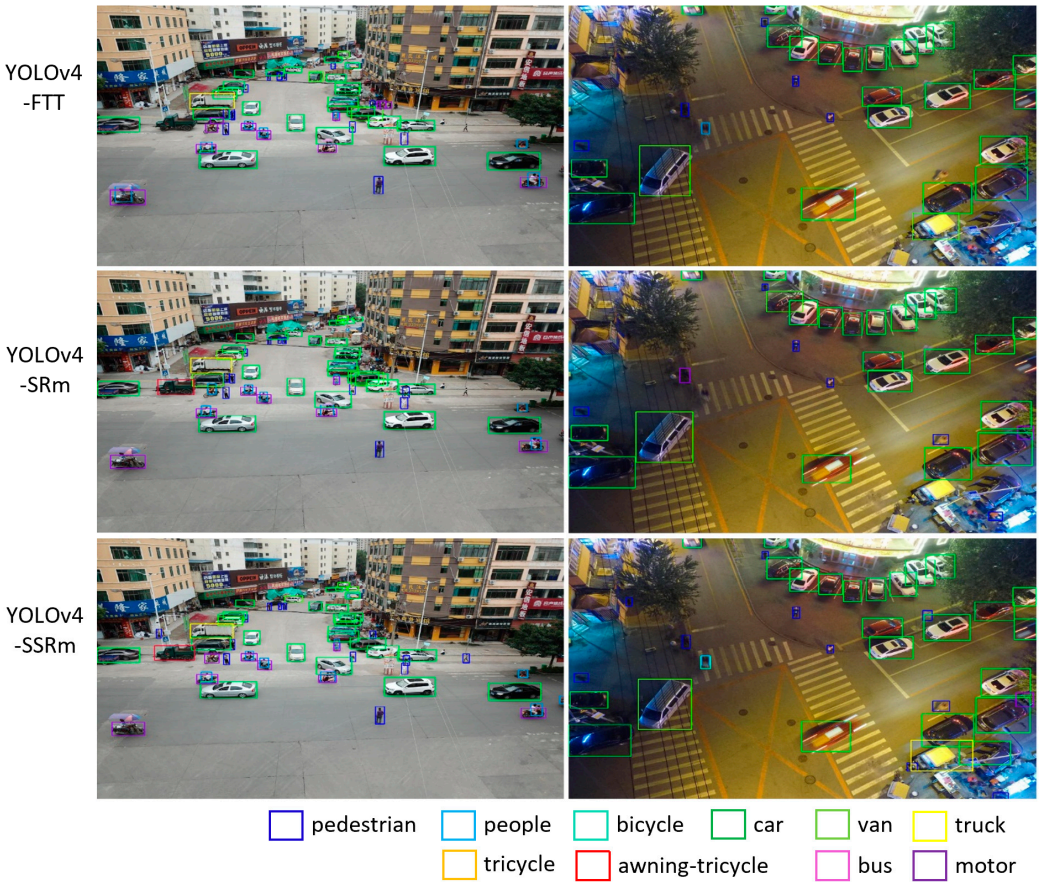

Figure 9. Detection results of the proposed methods on VisDrone2019 dataset images.

Table 4. Comparison of seven networks in terms of model size, computational cost, and inference time in the desktop environment.

Model	Params	Weight	Self-Built Dataset			VisDrone2019 Dataset		
			FLOPs	IT	FPS	FLOPs	IT	FPS
Vanilla YOLOv4	64.4	244	9.99	8.80	114	48.16	10.38	96
YOLOv8-L	43.6	165	11.56	9.43	106	55.78	10.05	99
SRGAN + YOLOv4	65.8	249	118.76	12.61	79	572.60	17.03	59
YOLOv4-HR	48.3	183	10.75	8.95	112	51.82	10.83	92
YOLOv4-FTT	55.0	208	20.29	9.16	109	97.83	11.14	90
YOLOv4-SRm	49.3	187	19.47	9.10	110	93.88	10.97	91
YOLOv4-SSRm	45.1	171	10.22	8.66	115	49.25	10.60	94

Params (M), Weight (MB), FLOPs (G), IT (inference time in ms).

Table 5. Comparison of six methods in terms of inference time in the embedded environment using the self-built dataset.

Model	Inference Time (ms)	FPS
Vanilla YOLOv4	31.17	32
YOLOv8-L	29.90	33
YOLOv4-HR	29.40	34
YOLOv4-FTT	40.02	25
YOLOv4-SRm	40.79	25
YOLOv4-SSRm	27.51	36

5. Conclusions

This paper proposes a novel combination of the SR technique and one-stage detector to address the limitations of predicting small objects in low-resolution images. We first suggest a way to use the FTT module as a part of the one-stage detector, and then propose a novel SR module and its simplified version. The proposed modules are adopted to the neck of the network to super-resolve feature maps and efficiently capture credible regional details of small objects using the pixel shuffling technique. Experiments with two datasets demonstrated the superiority of the proposed methods, especially in small object detection. In addition, this paper showed that the proposed methods can be implemented in practical real-time embedded systems with high frame rates. For future research, we will explore combining our modules with other state-of-the-art one-stage detectors and simplify the network based on channel pruning to increase the input image size for higher detection performance. We will also conduct more experiments on additional datasets to check whether the proposed method is effective in other applications.

Author Contributions: Conceptualization, X.N.H. and J.K.S.; methodology, X.N.H. and J.K.S.; software, X.N.H. and G.B.J.; validation, X.N.H., G.B.J. and J.K.S.; formal analysis, X.N.H., G.B.J. and J.K.S.; investigation, X.N.H.; resources, X.N.H. and J.K.S.; data curation, X.N.H. and G.B.J.; writing—original draft preparation, X.N.H.; writing—review and editing, J.K.S.; visualization, X.N.H.; supervision, J.K.S.; project administration, J.K.S.; funding acquisition, J.K.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2022R1F1A1074708), and in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2020R1A6A1A03038540).

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *Proc. IEEE Inst. Electr. Electron. Eng.* **2023**, *111*, 257–276. [CrossRef]
2. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
3. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the 28th Advances in Neural Information Processing Systems (NIPS'15), Montreal, QC, Canada, 7–10 December 2015; pp. 91–99.
4. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
5. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
6. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.

7. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
8. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
9. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Springer International Publishing: Cham, Switzerland, 2016; pp. 21–37.
10. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
11. Chen, G.; Wang, H.; Chen, K.; Li, Z.; Song, Z.; Liu, Y.; Chen, W.; Knoll, A. A Survey of the Four Pillars for Small Object Detection: Multiscale Representation, Contextual Information, Super-Resolution, and Region Proposal. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 936–953. [CrossRef]
12. Tong, K.; Wu, Y. Deep learning-based detection from the perspective of small or tiny objects: A survey. *Image Vis. Comput.* **2022**, *123*, 104471. [CrossRef]
13. Cheng, G.; Yuan, X.; Yao, X.; Yan, K.; Zeng, Q.; Xie, X.; Han, J. Towards Large-Scale Small Object Detection: Survey and Benchmarks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 13467–13488. [CrossRef] [PubMed]
14. Haris, M.; Shakhnarovich, G.; Ukita, N. Task-Driven Super Resolution: Object Detection in Low-Resolution Images. In Proceedings of the Neural Information Processing: 28th International Conference (ICONIP 2021), Sanur, Indonesia, 8–12 December 2021; pp. 387–395.
15. Pang, Y.; Cao, J.; Wang, J.; Han, J. JCS-Net: Joint Classification and Super-Resolution Network for Small-Scale Pedestrian Detection in Surveillance Images. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 3322–3331. [CrossRef]
16. Wang, Z.-Z.; Xie, K.; Zhang, X.-Y.; Chen, H.-Q.; Wen, C.; He, J.-B. Small-Object Detection Based on YOLO and Dense Block via Image Super-Resolution. *IEEE Access* **2021**, *9*, 56416–56429. [CrossRef]
17. Zhao, X.; Li, W.; Zhang, Y.; Feng, Z. Residual Super-Resolution Single Shot Network for Low-Resolution Object Detection. *IEEE Access* **2018**, *6*, 47780–47793. [CrossRef]
18. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the 27th Advances in Neural Information Processing Systems (NIPS'14), Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
19. Mostofa, M.; Ferdous, S.N.; Riggan, B.S.; Nasrabadi, N.M. Joint-SRVDNet: Joint Super Resolution and Vehicle Detection Network. *IEEE Access* **2020**, *8*, 82306–82319. [CrossRef]
20. Courtrai, L.; Pham, M.-T.; Lefèvre, S. Small Object Detection in Remote Sensing Images Based on Super-Resolution with Auxiliary Generative Adversarial Networks. *Remote Sens.* **2022**, *12*, 3152. [CrossRef]
21. Jin, Y.; Zhang, Y.; Cen, Y.; Li, Y.; Mladenovic, V.; Voronin, V. Pedestrian detection with super-resolution reconstruction for low-quality image. *Pattern Recognit.* **2021**, *115*, 107846. [CrossRef]
22. Bai, Y.; Zhang, Y.; Ding, M.; Ghanem, B. SOD-MTGAN: Small Object Detection via Multi-Task Generative Adversarial Network. In Proceedings of the Computer Vision—ECCV 2018: 15th European Conference, Munich, Germany, 8–14 September 2018; pp. 206–221.
23. Bai, Y.; Zhang, Y.; Ding, M.; Ghanem, B. Finding Tiny Faces in the Wild with Generative Adversarial Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 21–30.
24. Li, J.; Liang, X.; Wei, Y.; Xu, T.; Feng, J.; Yan, S. Perceptual Generative Adversarial Networks for Small Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1222–1230.
25. Noh, J.; Bae, W.; Lee, W.; Seo, J.; Kim, G. Better to Follow, Follow to Be Better: Towards Precise Supervision of Feature Super-Resolution for Small Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9725–9734.
26. Deng, C.; Wang, M.; Liu, L.; Liu, Y.; Jiang, Y. Extended Feature Pyramid Network for Small Object Detection. *IEEE Trans. Multimed.* **2021**, *24*, 1968–1979. [CrossRef]
27. Liu, Z.; Li, D.; Ge, S.S.; Tian, F. Small traffic sign detection from large image. *Appl. Intell.* **2019**, *50*, 1–13. [CrossRef]
28. Cui, L.; Ma, R.; Lv, P.; Jiang, X.; Gao, Z.; Zhou, B.; Xu, M. MDSSD: Multi-scale Deconvolutional Single Shot Detector for Small Objects. *arXiv* **2018**, arXiv:1805.07009. [CrossRef]
29. Zhang, G.; Lu, S.; Zhang, W. CAD-Net: A Context-Aware Detection Network for Objects in Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 10015–10024. [CrossRef]
30. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
31. Chen, C.; Ling, Q. Adaptive Convolution for Object Detection. *IEEE Trans. Multimed.* **2019**, *21*, 3205–3217. [CrossRef]
32. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv* **2015**, arXiv:1511.07122.
33. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In Proceedings of the 29th Advances in Neural Information Processing Systems (NIPS'16), Barcelona, Spain, 5–10 December 2016; pp. 379–387.

34. Wilms, C.; Frintrop, S. AttentionMask: Attentive, Efficient Object Proposal Generation Focusing on Small Objects. In Proceedings of the Computer Vision—ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, 2–6 December 2018; Revised Selected Papers. Springer International Publishing: Cham, Switzerland, 2019; pp. 678–694.
35. Chen, Z.; Wu, K.; Li, Y.; Wang, M.; Li, W. SSD-MSN: An Improved Multi-Scale Object Detection Network Based on SSD. *IEEE Access* **2019**, *7*, 80622–80632. [CrossRef]
36. Hu, P.; Ramanan, D. Finding Tiny Faces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 951–959.
37. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
38. Min, K.; Lee, G.-H.; Lee, S.-W. Attentional feature pyramid network for small object detection. *Neural Netw.* **2022**, *155*, 439–450. [CrossRef]
39. Bosquet, B.; Mucientes, M.; Brea, V.M. STDnet: Exploiting high resolution feature maps for small object detection. *Eng. Appl. Artif. Intell.* **2020**, *91*, 103615. [CrossRef]
40. YOLOv4. Available online: https://docs.nvidia.com/tao/tao-toolkit/text/object_detection/yolo_v4.html (accessed on 6 March 2023).
41. Getting Started with YOLO V4. Available online: https://www.mathworks.com/help/vision/ug/getting-started-with-yolo-v4.html (accessed on 6 March 2023).
42. Choi, K.; Wi, S.M.; Jung, H.G.; Suhr, J.K. Simplification of Deep Neural Network-Based Object Detector for Real-Time Edge Computing. *Sensors* **2023**, *23*, 3777. [CrossRef] [PubMed]
43. Du, D.; Zhu, P.; Wen, L.; Bian, X.; Lin, H.; Hu, Q.; Peng, T.; Zheng, J.; Wang, X.; Zhang, Y.; et al. VisDrone-DET2019: The Vision Meets Drone Object Detection in Image Challenge Results. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
44. Wang, J.; Yang, W.; Guo, H.; Zhang, R.; Xia, G.-S. Tiny Object Detection in Aerial Images. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 3791–3798.
45. YOLOv8. Available online: https://github.com/ultralytics/ultralytics (accessed on 2 January 2024).