IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 July 2024, accepted 16 July 2024, date of publication 23 July 2024, date of current version 1 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3432532

RESEARCH ARTICLE

Object Detection Using ESRGAN With a Sequential Transfer Learning on Remote Sensing Embedded Systems

YOGENDRA RAO MUSUNURI[1], CHANGWON KIM[2], OH-SEOL KWON[2], AND SUN-YUAN KUNG[3], (Life Fellow, IEEE)

[1]Department of Control and Instrumentation Engineering, Changwon National University, Changwon 51140, Republic of Korea
[2]School of Electrical, Electronics, and Control Engineering, Changwon National University, Changwon 51140, Republic of Korea
[3]School of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544, USA

Corresponding author: Oh-Seol Kwon (osk1@changwon.ac.kr)

This work was supported in part by the Regional Innovation Strategy (RIS) through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) under Grant 2021RIS-003; in part by the Development Project of Industry Academic-Research Platform Collaboration Technology in 2022 under Grant S3311946; and in part by the Ministry of Science and ICT (MSIT), South Korea, through the Information Technology Research Center (ITRC) Support Program Supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP) under Grant RS-2024-00438409.

ABSTRACT The field of remote sensing has experienced rapid advancement owing to the widespread utilization of image sensors, drones, and satellites for data collection. However, object detection in remote sensing poses challenges owing to small objects with low resolution (LR), complex scenes, and limited data for model training. Conventional methods rely on computationally intensive models and hardware setups that are not suitable for real-time detection. To address this issue, we propose a novel sequential transfer learning method based on generative adversarial networks (GANs) that generate super-resolved data from LR for embedded systems, enabling improved performance with limited data by combining learning from both heterogeneous and homogeneous data. Additionally, we train the model sequentially, starting with the easiest data and progressing to the most complex based on the complexity levels determined by the GAN-generated images. The GAN model is trained on a diverse dataset of images and learned to generate high-resolution images from the LR, capturing finer object details for enhanced accuracy and localization capabilities. The proposed method acquires more robust features and enhances the generalizability and convergence of the model. Furthermore, the trained model of the proposed method is deployed on embedded platforms, such as Nvidia's Jetson Nano and AGX Orin, for real-time remote-sensing object detection, with satisfactory detection performance. Evaluation metrics, such as mAP@0.5, mAP@0.5–0.95, and F1 score were used to assess the object detection accuracy. The experimental results demonstrated a significant improvement in accuracy when the proposed method was implemented with YOLOv7, achieving detection performance scores of 99.21, 98.57, 93.71, 78.38, 75.73, 48.68, 0.971, 0.971, and 0.911 on the VEDAI-VISIBLE, VEDAI-IR, and DOTA datasets, respectively.

INDEX TERMS Embedded system, Jetson AGX Orin, Jetson nano, real-time detection, remote sensing images, super-resolution, sequential transfer learning.

I. INTRODUCTION

Object detection is a crucial task in the field of computer vision, involving the precise detection, localization, and categorization of objects within an image or video frame. This technology finds applications in various fields, such as autonomous vehicles [1], [2], security systems [3], industrial robotics [4], medical imaging [5], and remote sensing [6], [7]. However, detecting small targets in aerial images poses a significant challenge owing to the low resolution (LR) of

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

background-like targets, which limits the available image information. Various techniques can be utilized to enhance object detection in LR images. Super-resolution (SR) [8], [9], [10], [11] and transfer learning (TL) [12], [13] are two such techniques that have shown promise in improving object detection accuracy.

SR is the process of increasing image resolution, generating the high-resolution images from an LR input while preserving as much detail as possible. These high-resolution images provide more accurate and detailed information for object detection, aiding in the identification and localization of small or blurry objects in LR images. In recent years, deep learning-based methods have addressed both tasks, with a growing interest in combining these approaches to enhance performance and efficiency. Techniques, such as SR have been leveraged to enhance input images for object detection, resulting in improved accuracy and robustness of detection algorithms. Mostofa et al. [14] utilized a multi-scale generative adversarial network (GAN) to learn hierarchical and discriminative features for small-object detection, whereas Musunuri et al. [15] implemented an object detection network to enhance object detection accuracy in deteriorated remote sensing images. Furthermore, Lian et al. [16] designed a network based on attention feature fusion to address small-object detection challenges in traffic scenes, to resolve the problem of overlapping image detection. Mu et al. [17] explored edge enhancement in SR to preserve edges in retrieved images, thereby improving small-object detection performance. The aforementioned methods exemplify the optimized SR techniques; however, they cannot effectively enhance object detection accuracy.

TL is a machine learning technique in which a model developed for one task is repurposed as the starting point for the model for a second task. TL, particularly for LR images, is effective when data or hardware resources are limited. Yan et al. [18] proposed a depth regression–based convolutional neural network (CNN) algorithm combined with TL to address the challenges of object detection stemming from varying object features. Hilal et al. [19] developed a model to resolve the semantic gap among various datasets using a deep transfer learning-based fusion model. Xu et al. [20] successfully implemented an effective TL for small-object detection to address the problem of limited training samples and low-quality images. Suseela and Kalimuthu [21] utilized TL-based SR in medical ultrasound images, Huang et al. [22] enhanced infrared images to bridge the gap between higher-dimensional feature spaces for improved model accuracy; and Talukdar et al. [23] employed TL in various object detection networks, including R-FCN [24], SSD [25], R-CNN [26], and Faster R-CNN [27], to enhance accuracy. However, these methods exhibited superior performance on high-resolution images compared with LR images, large datasets, and efficient resources. Therefore, to enhance detection accuracy with limited resources, this study introduces a novel approach.

This study proposes a sequential TL method based on the complexity of data utilized to train the learning model. This method has been applied to enhance the object detection accuracy for LR background-like targets in remote sensing images. The proposed method serves as a training strategy to enhance performance in scenarios of limited data through a combination of learning between heterogeneous and homogeneous data.

To prepare the data, a GAN-based SR model was utilized as a preprocessing module to enhance the images from LR to SR. These enhanced images were then organized based on difficulty level to facilitate training in a structured curriculum. Consequently, the proposed technique can acquire more robust features, thereby enhancing both the generalizability and convergence of the model.

In addition, the trained model resulting from the proposed method was deployed on embedded platforms, such as Nvidia's Jetson Nano and Jetson AGX Orin for real-time remote sensing object detection.

The remainder of this paper is organized as follows: Section II discusses related studies, the proposed method is examined in Section III, Section IV evaluates the effectiveness of the proposed method, and the conclusions of the study are presented in Section V.

II. RELATED WORKS

A. SR AND OBJECT DETECTION

Recently, the field of remote sensing has experienced rapid advancements owing to the utilization of image sensors, satellites, and drones. Object detection is crucial to locating objects from space and involves automatic detection and identification of objects in remote-sensing images. These images captured from aerial or satellite platforms offer a broad view of the Earth's surface. Object detection in remote sensing presents challenges, such as the presence of small-sized objects, complex image backgrounds, and difficulty in distinguishing objects from their surroundings. Variations in illumination conditions can impact the appearance of objects, whereas noise in images can further hinder the performance of object detection algorithms.

Despite these challenges, significant progress has been achieved in object detection in remote sensing owing to the development of deep learning techniques. Deep learning methods have proven effective in automatically identifying objects by extracting features from images, significantly enhancing the accuracy and robustness of object detection algorithms. Mu et al. [17] small-object detection in aerial images was explored to enhance the accuracy of LR images using super-resolution techniques. Yan [7] focused on aircraft detection by utilizing center-based proposal regions to achieve precise identification of aircraft. Courtrai et al. [28] investigated the challenge of detecting small objects in satellite images, elevating their performance through the integration of spatial super-resolution techniques. Chen et al. [29] developed a model based on YOLOv3 with an attention mechanism to address issues, such as cloud occlusions

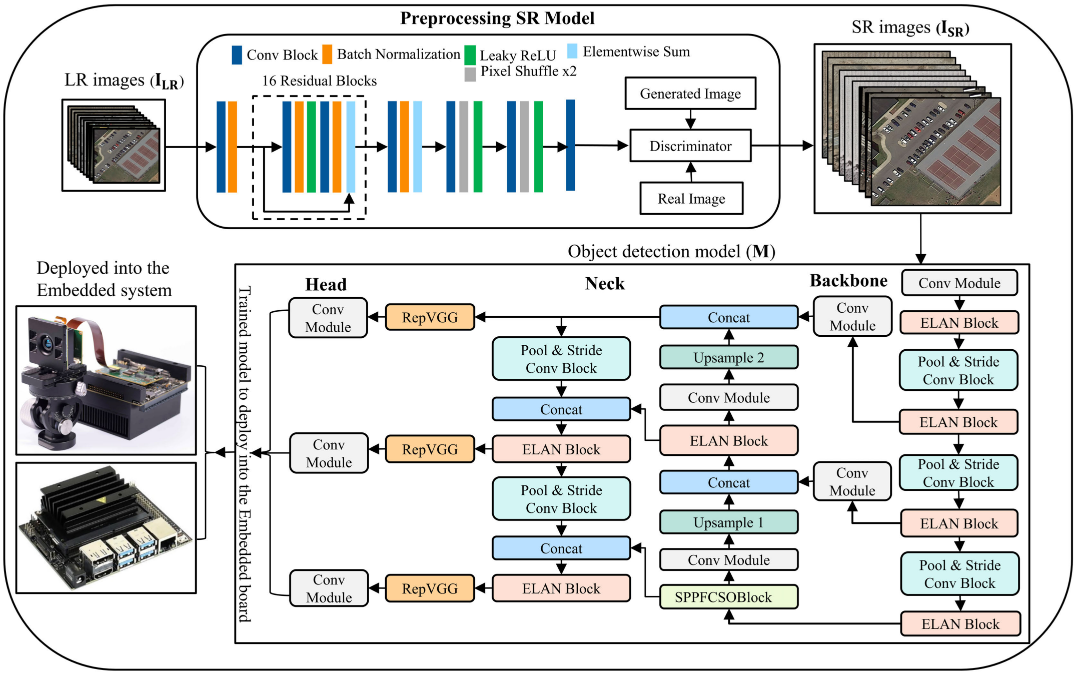


FIGURE 1. Flowchart of proposed method based on ESRGAN for embedded systems.

and strong waves, thereby enhancing the accuracy of ship detection. Wang et al. [30] developed a multiscale transformer fusion approach incorporating a convolutional block attention module for change detection.

Gong et al. [31] designed a context-aware CNN to enhance the detection accuracy under varying illumination intensities, weather conditions, and image quality in high-resolution remote sensing imagery. This network can simultaneously maintain high efficiency and effectiveness. Liu et al. [32] developed Road Net, a tool capable of automatically extracting road networks from high-resolution urban scene images. This tool predicts road surfaces, edges, and centerlines as well as addresses, such as shadows and occlusions along road signs. Tayara et al. [33] developed a convolution regression network specifically tailored for vehicle detection, considering complex backgrounds, such as trash bins, air conditioning units, and road marks.

Rabbi et al. [34] developed a model aimed at enhancing the accuracy of object detection, particularly for noisy LR images. Their method effectively addressed edge detection issues by employing an edge-enhanced network to recover edges, resulting in an enhanced image. Dong et al. [35] applied a CNN to scale object features and detect objects in spatial-resolution remote sensing images. This model improves the detection quality of various remote-sensing images. Rifat Arefin et al. [36] implemented a multi-image SR technique using recurrent networks, which are end-to-end deep neural networks based on the encoder-decoder model. This approach aims to reduce the costs associated with acquiring satellite imagery. Fernandez-Beltran et al. [37] proposed single-frame SR methods for analyzing remote sensing frameworks. Failure to properly restore SR models can have a negative impact the object detection performance. Thus, the proposed method has been applied to improve the accuracy of object detection in such cases.

B. TL IN OBJECT DETECTION

Deep learning models extract high-level features from vast amounts of data, making them more advanced than conventional machine learning techniques. However, data preparation can be costly and time-consuming owing to the complexity and quality of datasets with annotations. TL helps reduce the time required to create new datasets by leveraging knowledge from models trained on large datasets to train models on smaller datasets. Various types of TL methods exist [38], such as instance-based (which assigns appropriate weights to selected instances), mapping-based (which refers to mapping instances from the source to the target domain), network-based (which refers to reusing the partial network transfer to the target domain), and adversarial-based (which represents an adversarial transfer learning application cable to both the source and target domains). Pan et al. [39] developed a framework that combines cascaded convolutional neural networks with TL and geometric feature constraints for air detection, addressing the limitations of weakly supervised methods. However, these methods could not extract an adequate number of features, and their detection accuracy required improvement. In response to these challenges, Chen et al. [40] investigated a modified convolutional network with a TL to address issues, such as narrowing, sparsity, diversity, and class imbalance. Tong et al. [41] introduced a novel mathematical model into a new framework for quantifying transferability in multisource TL, utilizing deep learning networks for training. Liu et al. [42] utilized TL to enhance the performance of the YOLOv5 object detector for tassel detection, a task known for its difficulty because of the small size and complexity of objects of interest. By deriving center points from Center Net to improve bounding boxes and leveraging TL for additional reference data, significant improvements were achieved. Li et al. [43] utilized TL to provide a tradeoff between accuracy and speed of the inference of small targets using the YOLOv3 model. Their approach involved pruning to reduce network size, thereby enhancing detection accuracy and speed. Hong et al. [44] introduced the spectral GPT to address the research gap in spectral data, providing valuable insights for scene understanding in remote sensing applications. Li et al. [45] proposed the LRR-Net for hyperspectral anomaly detection, incorporating prior knowledge into deep networks to guide parameter optimization. Wu et al. [46] implemented the UIU-Net for infrared small-object detection to address challenges, such as tiny object loss and feature distinguishability with increasing network depth. In another study, [47], they designed an ORSIm detector for optical remote sensing imagery to tackle image deformations, such as objective scaling and rotation by integrating diverse channel feature extraction, feature learning, fast image pyramid matching, and boosting the strategy. Carion et al. [48] presented a detection transformer (DETR) to resolve the non-maximum suppression procedure or anchor generation problem, which explicitly encodes our prior knowledge about the task. Feng et al. [49] implemented task-aligned one-stage object detection (TOOD) to address spatial misalignments in predictions between two tasks.

TL is a powerful technique widely utilized in remote sensing and edge computing applications [50], [51], [52], [53] to enhance embedded image processing performance. In this context, a large dataset is typically a dataset of natural images containing millions of images labeled with object categories, whereas a smaller dataset is a dataset of remote sensing images labeled with object categories. Koshelev et al. [54] employed TL for drone-aided weed detection. The convolution network was trained to boost the segmentation performance and decrease computational cost. Athanasiadis et al. [55] implemented a framework for embedded systems using TL for object detection. This devised a mechanism for automatic hyperparameter optimization to accelerate the performance of the model for real-time applications using SqeezeDet. Gadiraju and Vatsavai [56] leveraged TL by utilizing pre-trained network weights as initial weights for training remote sensing datasets or freezing layers to enhance remote sensing classification performance on benchmark datasets. Ma et al. [57] developed a lightweight detector through model compression by applying sparsity, channel pruning, and layer pruning. This approach aims to optimize the edge-device-oriented target detection method by reducing computational complexity and memory consumption. Sun et al. [58] proposed the BiFA-YOLO method for ship detection in high-resolution SAR images. This method addresses the challenges posed by multi-scale, arbitrary directions, and dense arrangements, enabling quick and accurate ship detection. While many existing methods focus on TL, which involves the transfer of a pre-trained model, our research aims to overcome data scarcity, expert knowledge limitations, and resource constraints, making it a pivotal area of study. Hence, we propose a novel strategy for improving the accuracy of object detection in remote-sensing images.

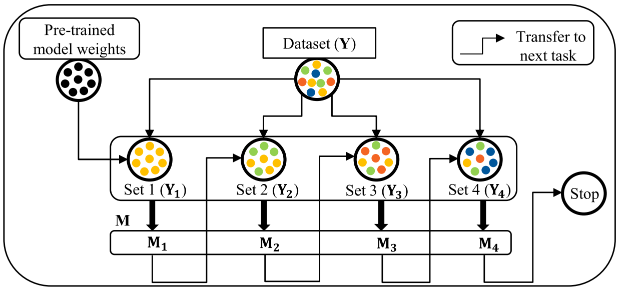


FIGURE 2. Proposed method to enhancing the accuracy of the object detection model.

III. PROPOSED METHOD

Detection of small objects in remote sensing images remains a challenging task. To address the challenge of detecting small objects in LR backgrounds, this study proposes a method in which the model is first optimized with SR images to enhance the LR images, followed by the application of a novel strategy to enhance object detection accuracy. The proposed model comprises three steps: an SR module utilizing a GAN and an object detection module (M), as shown in Fig. 1. The SR images generated in the first module were prepared as a dataset (Y) for training, and the proposed method was used in the second module, as shown in Fig. 2. M_1, M_2, M_3, and M_4 are the corresponding weights for transferring to the next task during the sequential process.

A. PROPOSED LEARNING METHOD

The proposed method is implemented on the optimized network to enhance accuracy, addressing the challenge of insufficient training data that can lead to information loss when attempting to retrieve information from SR-generated images. The proposed method, defined as the easy-to-hardest data training strategy, follows a curriculum-based approach. Subsequently, the model learns sequentially by transferring knowledge from easier to more challenging data, considering the complexity of the training data. To prepare the data, a GAN-based SR model served as a pre-processing module to enhance the images from LR to SR.

The SR module, based on an enhanced SR-GAN [59], generated SR images from the LR input. The key components of this module were the generator, discriminator, and loss functions. The generator utilizes the LR image as input and produces a high-resolution image, whereas the discriminator distinguishes between real high-resolution and SR images produced by the generator. A perceptual loss function is utilized to enhance the visual quality of images produced by the model. The image size of the LR input to the model was 128 × 128, which was then super-revolved to 512 × 512 on a scale four. The SR model presented in Fig. 1 and it is expressed using Eq. (1).

$$I_{SR} = H_{ESRGAN}\left(I_{LR}\right) \tag{1}$$

Here, H_{ESRGAN} is the convolution operation for feature extraction I_{LR} and output as I_{SR}.

To train the proposed method, the dataset **(Y)** was initially prepared with SR images sourced from publicly available datasets [14]. These SR images were then categorized from easiest to hardest data using conditions (a)–(c). **Y** was partitioned into **N** sub-sets $\mathbf{Y_1}$, $\mathbf{Y_2}$, $\mathbf{Y_3}$, $\mathbf{Y_4}$, ….., $\mathbf{Y_N}$, where **N** represents the total number of sets, with **N** increasing until the model's learning capacity reaches a minimum threshold. To designate the subsequent set as the target, the following requirements must be fulfilled: (a) A subsequent increase in the probability of sampling and diversity. The diversity is determined by the number of classes added to the next set. (b) The training samples were gradually updated, with the sample size of the training set increased, necessitating a corresponding increase in the weight of the next target set. (c) All samples within each set were accumulated to form the final set, ensuring uniformity among all samples. These conditions outline the transfer of the proposed strategy to the next target set for model training and updates.

Each set was categorized based on its diversity, structural complexity, and accumulation of images from one set to another for VEDAI-VISIBLE [14], [15], VEDAI-IR [14], [15], and DOTA [14], [15]. The number of categories varied across datasets. VEDAI-VISIBLE and IR consisted of two categories: cars and trucks. In contrast, the DOTA 1.0 dataset included 15 categories: plane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, swimming pool, large vehicle, small vehicle, helicopter, roundabout, and soccer ball field. For the DOTA experiment, we focused only on two categories—cars and trucks—related to vehicles. Labels were manually annotated using the annotated software. Structural complexity was determined by the number of classes within a single image: easy for less than two classes, moderate for less than four classes, and high for more than four classes. Diversity was achieved through image augmentation techniques, generating new samples from existing ones.

Set $\mathbf{Y_1}$*:* This serves as the initial set, comprising 10% of samples from the available dataset of 600 VEDAI-VISIBLE images of autonomous vehicles, totaling 60 samples. The initial set comprised 60 images with two classes and their corresponding labels, exhibiting a low structural complexity with no image augmentation.

Set $\mathbf{Y_2}$*:* This is the second set prepared using the number of classes and image augmentation. This set contained 60 images with their corresponding labels and augmented images, such as image rotations at 90°, 180°, and 270°. The total number of images was 240 with corresponding labels and moderate structural complexity.

Set $\mathbf{Y_3}$*:* This is the third set prepared using the number of classes and image augmentation. It comprises 60 images with labels and augmented images, such as image cropping, horizontal flipping, vertical flipping, and horizontal and vertical flipping, resulting in a total of 300 images with high structural complexity.

Set $\mathbf{Y_4}$*:* This is the fourth set prepared by accumulating $\mathbf{Y_1}$, $\mathbf{Y_2}$, and $\mathbf{Y_3}$. This set includes augmented images from rotations at 90°, 180°, and 270°, as image cropping, horizontal flipping, vertical flipping, and horizontal and vertical flipping. The total number of images was 600, with high structural complexity.

The training process is described as follows: (a) First, Set $\mathbf{Y_1}$ was trained with the pre-trained COCO [60] weights to fine-tune the training dataset (Y). (b) The pre-trained weights from Set $\mathbf{Y_1}$ with TL were utilized to train Set $\mathbf{Y_2}$. TL is the process of transferring knowledge from a pre-trained model to address a related task. Among the various TL techniques available, a fine-tuning method was utilized to modify pre-existing model parameters for a new task utilizing a limited amount of labeled data to prevent overfitting. The proposed method can be explained as follows. After the initial training on Set $\mathbf{Y_1}$, the model leveraged knowledge from Set $\mathbf{Y_1}$ to train Set $\mathbf{Y_2}$. (c) Set $\mathbf{Y_3}$ was then trained using the weights of sets $\mathbf{Y_2}$ and (d). Set $\mathbf{Y_4}$ was trained using all the knowledge of the model up to Set $\mathbf{Y_3}$. This sequential process continues until the model achieves a minimum value of accuracy improvement, at which point the process is halted. The procedure for the proposed method is described below.

Algorithm: Proposed Method for Object Detection

```
Input: Y: training dataset; C: complexity; T: transfer weights
Output: M: Optimal model to improve the accuracy until max.
$Y' = sort\,(Y, C)$
$\{Y_1, Y_2, \ldots, Y_k\} = Y'\ where\ C\,(y_a) < C\,(y_b)\,, y_a \in Y_i,\ \ y_b \in Y_j, \forall i < j$;
$Y^{train} = \emptyset$;
for $n = 1 \ldots k$ do
    $Y^{train} = Y^{train} \cup T^{n-1}$;
    while not improve the accuracy for $p$ epochs do
        $train\left(M, Y^{train}\right)$
    end while
end for
```

B. OBJECT DETECTION MODEL (M)

The object detection module, an advanced version of YOLO [60], is designed to enhance speed and accuracy by utilizing a single neural network to predict bounding boxes and class probabilities of objects in real-time images. The key components of this module include feature extraction and fusion layers, model scaling for concatenation, and a prediction head to enhance the accuracy of the LR remote-sensing images. The feature extraction layer employs an extended efficient layer aggregation network to extract features and uses them to expand, shuffle, and merge for enhanced learning ability. The model is then scaled to accommodate different inference speeds for real-time object detection.

Finally, the head predicts the object classes in the image, and optimization is conducted to enhance the image quality and retrieve more accurate image information, thereby improving the accuracy of the LR remote-sensing images, as shown in Eq. (2).

$$Y_{Output} = H_{YOLO}\left(I_{SR}\right) \tag{2}$$

Here, H_{YOLO} is the network used to classify the objects in the SR images I_{SR}.

C. LOSSES OF THE NETWORK

Finally, network loss occurs during the training. The multiple loss functions are adversarial, perceptual and the detection loss functions are expressed in the eq.(3), (4), and (5). The adversarial loss is L_{adv} defined as the probability of discriminator $D\left(G\left(I^{LR}\right)\right)$ that reconstructed image $G\left(I^{LR}\right)$ is a real high-resolution image.

$$\begin{aligned}L_{adv} = \underset{G}{min}\ \underset{D}{max} &\left[E_{I^{HR}\sim P_{train}}\left(I^{HR}\right)\left[logD\left(I^{HR}\right)\right]\right]\\ &+ E_{I^{LR}\sim P_G}\left(I^{LR}\right)\left[\left[log\left(1-D\left(G\left(I^{LR}\right)\right)\right)\right]\right]\end{aligned} \tag{3}$$

Here $P_{train}\left(I^{HR}\right)$ and $P_G\left(I^{LR}\right)$ define the probability distribution of real high-resolution images with respect to their LR images. Furthermore, the perceptual loss of the network during training, as represented by eq. (4).

$$\begin{aligned}L_{per} = &\frac{1}{N}\sum_{n=1}^{N}\frac{1}{C_jW_jH_j}\\ &\times\sum_{c=1}^{C_j}\sum_{x=1}^{W_j}\sum_{y=1}^{H_j}\left(\text{Ø}_j\left(I_n^{HR}\right)_{c,x,y} - \text{Ø}_j\left(G\left(I_n^{LR}\right)\right)_{c,x,y}\right)^2\end{aligned} \tag{4}$$

Here Ø_j is the j^{th} convolution layer feature map, and C_j, W_j and H_j are the feature maps of the network. The total loss of the network is defined as the summation of adversarial, and perceptual losses to optimize the network to retrieve the high-frequency details for better object detection in the LR remote sensing images. The loss function of the object detection model is shown in the eq. (5). It combines localization loss, confidence loss, and classification loss.

$$\begin{aligned}L_{detection} = &\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left(x_i-\hat{x}_i\right)^2+\left(y_i-\hat{y}_i\right)^2\\ &+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2\\ &+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2+\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}{}^{obj}l\left(C_i,\hat{C}_i\right)\\ &+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}{}^{obj}l\left(C_i,\hat{C}_i\right)\\ &+\sum_{i=0}^{S^2}1_i^{obj}\sum_{c\in classes}l\left(p_i\left(c\right)-\hat{p}_i\left(c\right)\right)\end{aligned} \tag{5}$$

Herein 1_{ij}^{obj} is the object detected by the j^{th} boundary box of grid cell i. x_i, y_i, w_i, h_i are the actual bounding box coordinates and predicted bounding box coordinates are $\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i$. C_i is the confidence score of actual box in cell i, $\hat{C}_i$ is confidence score of the predicted box.

IV. EXPERIMENT RESULTS AND DISCUSSION

A. DATASETS TO PREPARE THE TRAINING DATA AND METRICS

In the experiments, three public remote-sensing datasets for object detection were utilized: VEDAI-VISIBLE [14], [15], VEDAI-IR [14], [15], and DOTA [14], [15]. The first two datasets consisted of 1210 images with resolutions of 1024 × 1024 and 512 × 512. The third dataset (DOTA 1.0) comprised 2806 images captured by different sensors of various resolutions ranging from 800 × 800 to 20,000 × 20,000. The object detection performance was evaluated using mean average precision (mAP) @ 0.5, @0.5–0.95, and F1 metrics to evaluate the object detection performance. To train the proposed method, we divided the datasets into a training set (70%) and validation set (30%).

TABLE 1. Configuration details of experimental hardware and softwares.

S. No	Name of the Module	Specification
1	CPU	Intel Xenon Silver 4214R
2	RAM	512 GB
3	GPU	NVIDIA RTX A6000
4	Operating System	Windows 10 Pro.10.0.19042, 64 bit
5	CUDA	CUDA 11.2 with Cudnn 8.1.0
6	Data Processing	Python 3.9, OpenCV 4.0
7	DL Frame work	PyTorch 1.7.0

B. HARDWARE DETAILS AND TRAINING PARAMETERS

The experimental procedure and results of the proposed method are outlined in this section. All models were trained and tested on a single deep learning computer equipped with an NVIDIA RTX A6000 graphics card and CUDA. Experiments were conducted on the VEDAI-VISIBLE, VEDAI-IR, and DOTA datasets, with the configuration details of the experimental equipment listed in Table 1. The training parameters are as follows: the optimizer was a stochastic gradient descent algorithm used to minimize the loss function, with momentum, weight decay function, initial learning rate, batch size, intersection of union (IoU) threshold, epochs, and input image size set at 0.937, 0.0005, 0.01, 16, 0.25, 300, and 512 × 512, respectively. The confidence threshold for the prediction box was set to 0.001 and the IoU threshold for non-maximum suppression (NMS) was set to 0.65.

C. QUANTITATIVE RESULTS OF THE PROPOSED METHOD WHILE IMPLEMENTING ON OBJECT DETECTION MODELS

In this section, we present the quantitative results obtained by applying the proposed method to various

TABLE 2. Comparison of detection performance of the proposed and existing state-of-the-art methods.

S. No	Name of the Model		VEDAI-VISIBLE [14] [15]		VEDAI-IR [14] [15]		DOTA [14] [15]	
			mAP @ 0.5	F1 Score	mAP @ 0.5	F1 Score	mAP @ 0.5	F1 Score
1	YOLOv3_SRGAN [14]		62.45	0.591	70.10	0.687	86.18	0.837
2	YOLOv3_MSRGAN [14]		66.74	0.643	74.61	0.723	87.02	0.859
3	YOLOv3_EDSR [15]		74.32	0.754	70.62	0.727	91.47	0.889
4	YOLOv5_ESRGAN [42] [59]		59.22	0.549	69.43	0.632	71.70	0.701
5	YOLOv7_ESRGAN [59] [60]		46.42	0.495	73.65	0.6945	69.55	0.686
6	Detectron2_ESRGAN [59] [61] [62]		53.14	0.670	34.04	0.487	34.62	0.459
7	YOLOv8_ESRGAN [59] [63]		97.46	0.926	98.12	0.955	90.80	0.867
8	Proposed Method on	Detectron2_ESRGAN	60.77	0.738	54.55	0.693	77.20	0.846
9		YOLOv3_ESRGAN	99.10	0.978	99.25	0.985	98.48	0.964
10		YOLOv5_ESRGAN	98.36	0.951	99.03	0.969	97.01	0.941
11		YOLOv7_ESRGAN	99.21	0.971	98.57	0.971	93.71	0.911
12		YOLOv8_ESRGAN	99.06	0.969	99.42	0.988	98.05	0.956

TABLE 3. Comparison of accuracy of pre-and post-proposed method.

Name of the Model	VEDAI-VS [14]	VEDAI-IR [14]	DOTA [14]
	mAP @ 0.5-0.95	mAP @ 0.5-0.95	mAP @ 0.5-0.95
Pre-proposed method			
YOLOv3_ESRGAN	62.09	68.11	63.32
YOLOv5_ESRGAN	22.66	32.15	27.95
YOLOv7_ESRGAN	17.11	35.04	28.62
YOLOv8_ESRGAN	60.51	66.08	57.73
Post-proposed method			
YOLOv3_ESRGAN	70.15	71.98	56.95
YOLOv5_ESRGAN	60.85	63.34	57.02
YOLOv7_ESRGAN	78.38	75.73	48.68
YOLOv8_ESRGAN	74.27	79.76	67.46

state-of-the-art object detection models are listed in Table 2. The evaluation focused on assessing the effectiveness of the proposed method in enhancing model performance. In addition, the proposed method was utilized to compare the detection outcomes of YOLO-and SR-optimized networks, such as YOLOv3_SRGAN [14], YOLOV3_MSRGAN [14], YOLOv3_EDSR [15], YOLOv5_ESRGAN [42], [59], YOLOv7_ESRGAN [59], [60], Detectron2_ESRGAN [59], [61], and YOLOv8_ESRGAN [59], [63]. Detectron2 [61] is a modular and flexible object detection library developed by Facebook AI Research. Retina Net [62] is a backbone that effectively addresses class imbalance during the training of dense detectors by utilizing focal loss. This method is widely applied in object detection, instance segmentation, and panoptic segmentation. While the existing models listed in Table 2 primarily focus on SR-based object detection, they often fail to retain crucial information and do not yield significant improvements in accuracy. The proposed approach proves highly effective in scenarios in which expert knowledge is lacking, limited labeled data, or constraints in hardware and software resources.

Under such conditions, the proposed method enhanced the performance of super-resolved LR remote-sensing images. We compared our findings with those of ESRGAN as Detectron2, YOLOv3, v5, v7, and v8. The results presented in Table 2 revealed that the proposed method outperformed the others and resulted in mAP@0.5 on YOLOv8_ESRGAN and YOLOv7_ESRGAN (99.06, 99.21, 99.42, 98.57, 98.05, and 93.71), and F1 scores (0.969, 0.971, 0.988, 0.971, 0.956, and 0.911, respectively). When compared with the baseline models YOLOv7_ESRGAN and YOLOv8_ESRGAN, the proposed method demonstrated significant improvements of 36% and 1.6% with the VEDAI-VISIBLE data, 20% and 1.3% with the VEDAI-IR data, and 13% and 8.43% with the DOTA data, respectively. In addition to YOLO, the proposed method on Detectron2 was mAP@0.5 (60.77, 54.55, and 77.20), and the F1 score (0.738, 0.693, and 0.846) evenly increased the accuracy by 7.63%, 20.51%, and 42.58% on three datasets, respectively. The backbone of Detectron2 was trained with the focal loss, and it aligned with the speed of previous one-stage detectors and improved the accuracy.

To assess the efficacy of the proposed method in comparison to state-of-the-art techniques, we conducted an evaluation using the mAP@0.5-0.95 metric, as shown in Table 3. The experimental results demonstrated the effectiveness of the proposed method in enhancing the performance of various models along with the ESRGAN, such as YOLOv3, v5, v7, and v8. Prior to the implementation of our method, the mAP@0.5–0.95 values were 62.09, 22.66, 17.11, 60.51, 68.11, 32.15, 35.04, 66.08, 63.32, 27.95, 28.62, and 57.73. Following the integration of our method, notable improvements were observed, with mAP@0.5–0.95 values of 70.15, 60.85, 78.38, 74.27, 71.98, 63.34, 75.73, 79.76, 56.95, 57.02, 48.68, and 67.46 for VEDAI_VS, IR, and DOTA datasets, respectively. Comparatively, YOLOv5 and v7 demonstrated greater accuracy compared with YOLOv3, and v8, with YOLOv8 exhibiting the most significant improvement in the VEDAI-IR and DOTA datasets, whereas YOLOv7 exhibited the most significant improvement in the VEDAI-VISIBLE dataset. The red text in the results indicates the best-performing models, whereas the green text signifies the second-best performers.

Furthermore, we explored other object detection methods, such as UIU-Net [46], ORSIm detector [47], DETR [48], and TOOD [49]. While YOLO models and Detectron2 serve as versatile and robust general-purpose object detectors, specialized models, such as UIU-Net [46] and ORSIm detectors [47] offer superior performance in particular applications. DETR introduces a novel transformer-based approach that simplifies the detection pipeline and enhances performance in complex scenarios, whereas TOOD provides an efficient one-stage detection solution with enhanced task alignment. Each model possesses unique strengths that cater to diverse use cases and application requirements. UIU-Net and ORSIm demonstrated advancements in their specialized domains, focusing on infrared small-object detection and optical remote-sensing imagery. DETR showcases robust performance due to its transformer-based architecture, albeit with increased inference times. TOOD offers a balanced improvement in accuracy and efficiency, although it may require more computational resources and inference time compared with YOLO.

All models demonstrated a significant enhancement in mAP@0.5, 0.5–0.95, and F1 scores when the proposed method was implemented. However, YOLOv7 and v8 exhibited the highest performance gains in both mAP metrics and F1 scores. The proposed method is easily adaptable to YOLO owing to its simplicity, generalized models, pre-trained models, and seamless deployment into embedded systems, such as Jetson AGX Orin. For embedded object detection, achieving a tradeoff between accuracy and speed of inference is crucial. Therefore, the proposed method can enhance the object detection capabilities of remote-sensing object-detection networks with limited labeled data.

D. COMPARING THE PROPOSED METHOD ON LR, SR AND GT

The primary objective of the proposed model is shown in Fig. 3, illustrating the distinction between LR and SR alongside the ground truth (GT). In LR images, objects appear unclear or blurry, whereas the quality of SR images is enhanced such that objects are visible. The object detection accuracies of LR, SR, and GT, compared with those of both TL and the proposed strategy, are shown in Fig. 3. The detection performance was notably poor for LR images, prompting the development of an SR-based object detection network with the proposed method.

E. ABLATION STUDIES

To assess the impact of each set in the proposed method, we progressively increased the difficulty of data and compared the differences between the sets (refer to Tables 4 and 5). Based on experimental results, this strategy is advantageous for enhancing model performance. The optimal performance was observed in the final set because of the sequential model learning. The proposed method involved adjustments to network parameters, learning rate, and batch size to optimize model performance. Our analysis indicates that the number of sets required based on the model's learning ability, is equal to the minimum value of the experimental data, with the specific number varying based on the type of data utilized. The performances of each set are listed in Tables 4 and 5. Based on the experimental data, four sets were required for both VEDAI-VISIBLE and VEDAI-IR. However, more than four sets were required for DOTA to reach the minimum value rather than improve the performance because the model converged faster on VEDAI-VISIBLE and VEDAI-IR than on DOTA.

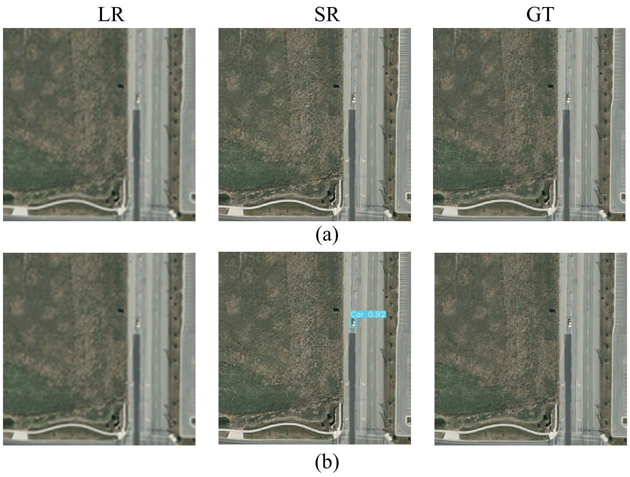


FIGURE 3. Comparison of LR, SR, and GT. (a). Conventional TL, (b). After applying the proposed method.

TABLE 4. Comparison of accuracy of each set of the proposed method on VEDAI-VS and VEDAI-IR.

Proposed Method	VEDAI-VS [14] [15]		VEDAI-IR [14] [15]	
	mAP @ 0.5	F1 Score	mAP @ 0.5	F1 Score
Set Y_1	62.50	0.617	78.13	0.738
Set Y_2	91.63	0.875	96.19	0.933
Set Y_3	97.77	0.938	97.48	0.940
Set Y_4	99.21	0.971	98.57	0.971

TABLE 5. Comparison of accuracy of each set of the proposed method on the DOTA dataset.

S. No	Proposed Method	DOTA [14] [15]	
		mAP @ 0.5	F1 Score
1	Set Y_1	80.27	0.769
2	Set Y_2	88.08	0.847
3	Set Y_3	91.63	0.888
4	Set Y_4	93.71	0.911

The performances of each set using the proposed method are listed in Tables 4 and 5. As described in Section II, the initial performance metrics for Set $\mathbf{Y_1}$ in terms of mAP@0.5 and F1 values were 15.40, 0.250, 12.23, 0.210, 15.61, and 0.253, respectively. After incorporating pre-trained model weights, the accuracy improved significantly, allowing the model to learn from its previous training. In particular, set $\mathbf{Y_1}$_TL weights were transferred to Set $\mathbf{Y_2}$, resulting in performances of 62.50, 0.617, 78.13, 0.738, 80.27, and 0.769. Set $\mathbf{Y_2}$ weights were then applied to Set $\mathbf{Y_3}$, where the model learned again, with accuracies of 97.77, 0.938, 97.48, 0.940, 91.63, and 0.888. Set $\mathbf{Y_3}$ weights were transferred to Set $\mathbf{Y_4}$, which achieved near-maximum accuracy, indicating that the model

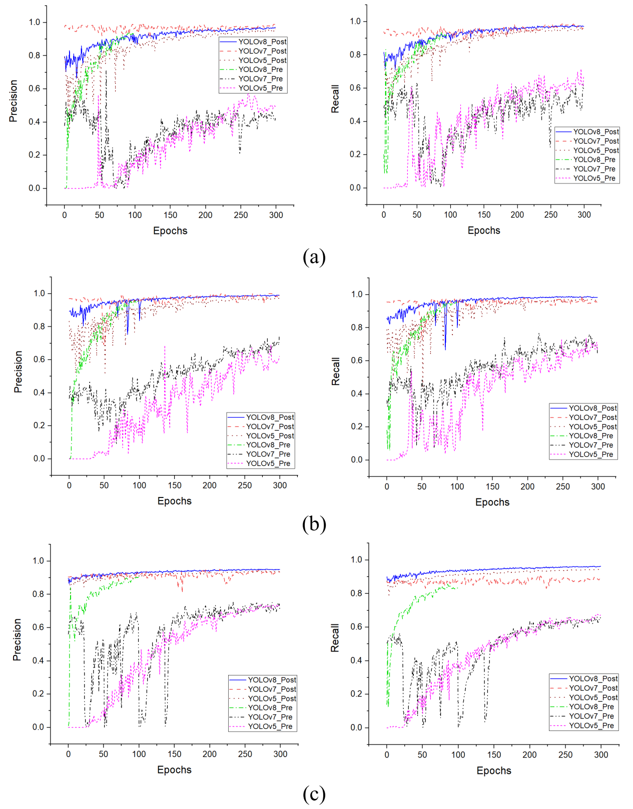


FIGURE 4. Precision and recall curves for various datasets, such as (a). VEDAI-VISIBLE, (b). VEDAI-IR, and (c). DOTA.

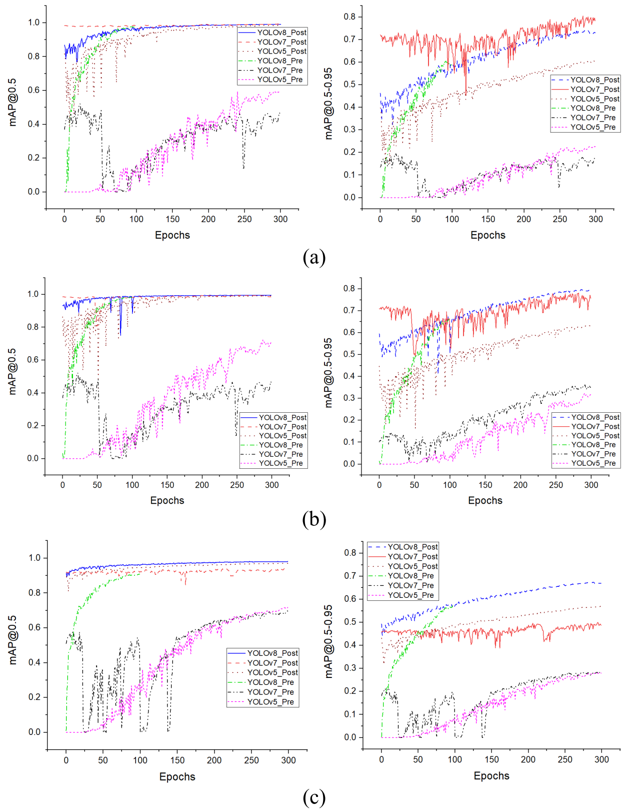


FIGURE 5. mAP@0.5 and mAP@0.5-0.95 curves on various datasets, such as (a). VEDAI-VISIBLE, (b). VEDAI-IR, and (c). DOTA.

stopped learning when the proposed strategy was used at 99.21, 0.971, 98.57, 0.971, 93.71, and 0.911. The cumulative performance of the proposed method is summarized in Table 2, showcasing higher accuracy in object detection and localization, particularly in challenging scenarios involving low-resolution or degraded images. Learning increased by 58% for VEDAI-VISIBLE data, and 66% for VEDAI-IR data after the pre-trained weights were transferred to set $\mathbf{Y_1}$. Once the maximum accuracy was reached, learning increased by 2% for both VEDAI-VISIBLE and VEDAI-IR datasets. In DOTA, the initial learning increased by 65% owing to the challenging and varied nature of the image data. In comparison, the accuracy peaked for both VEDAI-VISIBLE and VEDAI-IR datasets. To analyze the three datasets, we conducted comparisons across the four sets as DOTA did not achieve maximum accuracy. According to Tables 3 and 4, the proposed method is a solution for improving accuracy.

The experimental analyses delve into data using different datasets from the pre- and post-implementation of our method. A performance comparison of the three datasets: VEDAI-VISIBLE, VEDAI-IR, and DOTA is shown in Figs. 4 and 5. The precision and recall performances on visible and infrared data are shown in Figs. 4(a)–(c). Visible images captured in the light spectrum perceptible to the human eye offered rich color information crucial for object differentiation and scene comprehension. Infrared images capture the heat emitted by objects, allowing for visual capabilities in low-light or nighttime conditions, proving invaluable in surveillance, search and rescue, and military applications. Thus, the precisions of the proposed method for the three datasets were 0.9629, 0.9882, and 0.9339 for YOLOv7 and 0.9660, 0.9896, and 0.9490 for YOLOv8. The corresponding recall values were 0.9795, 0.9543, 0.8899 and 0.9730, 0.9862, 0.9624. The mAP@0.5 and 0.5–0.95 performance on YOLOv5, v7, and v8 are shown in Figs. 5(a)–(c). A comparison between pre- and post-proposed methods is detailed in Tables 2 and 3. The proposed method was trained on SR-augmented images to enhance the diversity of data for LR images.

However, during image reconstruction, the SR model lost edge information compared with the other models. This could potentially impact the model's performance during training. Despite this, the proposed method learned from the data and performed optimally. As shown in Figs. 4 and 5, the precision, recall, and mAP were outperformed when compared with the results of methods that do not incorporate the proposed method, such as the YOLO models optimized with the SR model. Most existing methods measure performance using mAP@0.5. As mentioned in YOLOv8, the proposed method was compared using mAP@0.5–0.95. The accuracy of the proposed method increased with the mAP@0.5–0.95 metric on detectron2, YOLOv3, v5, v7, and YOLOv8.

TABLE 6. Configuration of embedded system.

S. No	Name of the Module	Jetson Nano	Jetson AGX Orin
1	Features Size	70 × 45 mm	100 × 87 mm
2	Installed Package	Jetpack 4.6.1 [L4T 32.7.1]	Jetpack 5.1 [L4T 35.2.1]
3	HW Accelerator	128-core NVIDIA Maxwell GPU	2048-core NVIDIA Ampere GPU with 64 Tensor cores and 2× NVDLA Engines
4	Processor	Quad-Core ARM Cortex-A57	12-Core ARM Cortex-A78
5	Memory	4GB LPDDR4	64GB LPDDR5
6	Storage	16 GB eMMC flash	64 GB eMMC 5.1
7	Peak performance	472 GFLOPS	275 TOPS
8	Native Precision support	FP 16/FP 32	FP 16/FP 32
9	Nominal power	5/10 W	15/30/50 W
10	Weight	140g	880g
11	Operating System	Ubuntu 18.04 LTS, 64 bit	Ubuntu 20.04.06 LTS, 64 bit
12	Data Processing	Python 3.8, OpenCV 4.5.1	Python 3.8, OpenCV 4.5.4
13	DL Frame work	PyTorch 1.8.0, Torch vision:0.9.0	PyTorch 2.0.0, Torch vision:0.15.0

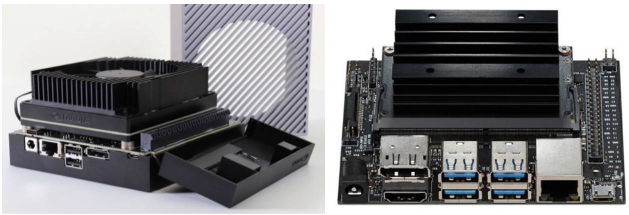

(a) (b)

FIGURE 6. Deploying on the embedded system, (a). NVIDIA Jetson AGX Orin, and (b). Jetson Nano.

F. DEPLOYING THE TRAINED MODEL INTO THE EMBEDDED SYSTEM FOR REMOTE SENSING OBJECT DETECTION

Remote sensing involves acquiring data and imagery from sensors on various platforms, such as satellites, airplanes, or drones. In this context, object detection refers to identifying and categorizing objects within these images.

Implementing object detection on embedded systems presents challenges and considerations owing to resource constraints, such as the selection of embedded platforms, dataset preparation, model selection, training, optimization, deployment, testing, and evaluation. Based on these parameters, we selected Nvidia's Jetson AGX Orin and Jetson Nano for our target applications, as shown in Fig. 6. The configuration details of the embedded system are listed in Table 6. AGX Orin was released in the spring of 2023 and is currently the most powerful available Jetson board. It features an Ampere GPU microarchitecture with 64 Tensor Cores and two NVDLA (NVIDIA deep learning accelerator) engines. These chips have been designed to efficiently perform standard neural network operations, such as convolutions. Thus, the overall peak performance of AGX Orin is approximately 275 TOPS. Jetson Nano, released in June 2019, was tailored for applications where reducing board size, power consumption, and price are crucial factors. The hardware acceleration featured an NVIDIA Maxwell GPU with a peak performance of 472 GFLOPs. Therefore, it does not utilize tensor cores or NVDLA engines for inference acceleration. Furthermore, these boards offer a cost-effective and efficient solution for a wide range of vision-based tasks, such as image classification, object detection, and segmentation. While embedded boards may have limited computational capabilities, they are sufficient for executing inference computations in real time but not for training models. The training of the proposed method was not conducted on the embedded system but performed on a deep learning workstation, as shown in Table 1, owing to the significant computational resources required compared with the inference process. Once the model was trained and weights were obtained, it was deployed on the target hardware for execution.

When the proposed method was deployed in an embedded system, the model comprised 314 layers, 36487166 parameters, and 6M gradients. The proposed method effectively enhanced the accuracy; however, it was less effective in reducing the network complexity. The deployed system could perform up to 275 TOPS in real time. The inference and NMS times were employed as real-time metrics to measure the speed of the model, as listed in Tables 7 and 8.

TABLE 7. Performance of trained networks in simulations performed on NVIDIA Jetson Nano embedded system.

	NVIDIA Jetson Nano		
Name of the dataset	VEDAI		DOTA
	VISIBLE	IR	
Average run inference time (sec)	0.7867	0.7847	1.3458
Average NMS time (sec)	0.0239	0.0412	0.2143

TABLE 8. Performance of trained networks in simulations performed on Jetson AGX Orin embedded system.

	NVIDIA AGX Orin		
Name of the dataset	VEDAI		DOTA
	VISIBLE	IR	
Average run inference time (sec)	0.07879	0.07276	0.07585
Average NMS time (sec)	0.00371	0.00370	0.00391

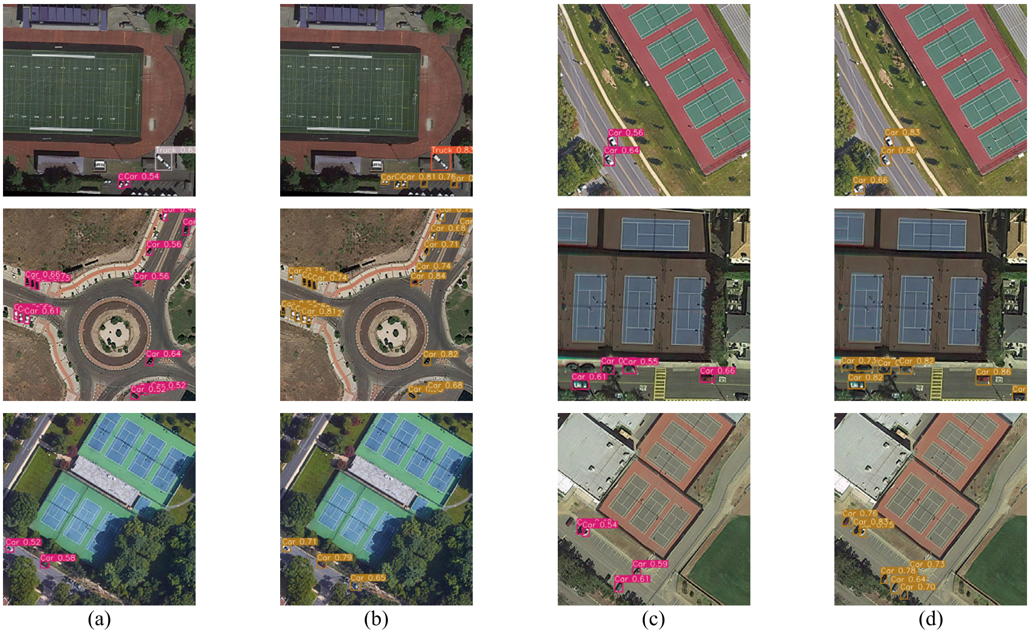


FIGURE 7. Detection results on embedded system for DOTA: (a), (c). conventional TL, (b), (d). Proposed method.

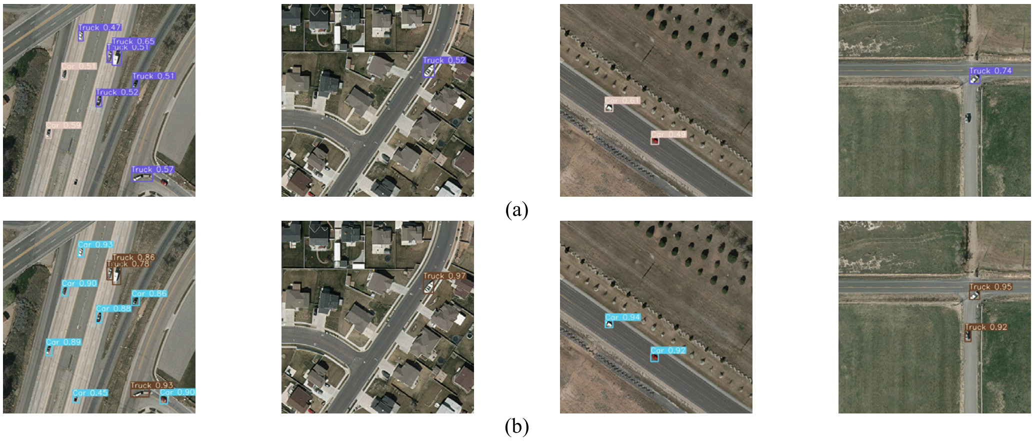


FIGURE 8. Detection results on embedded system for VEDAI-VISIBLE, (a). conventional TL, (b). Proposed method.

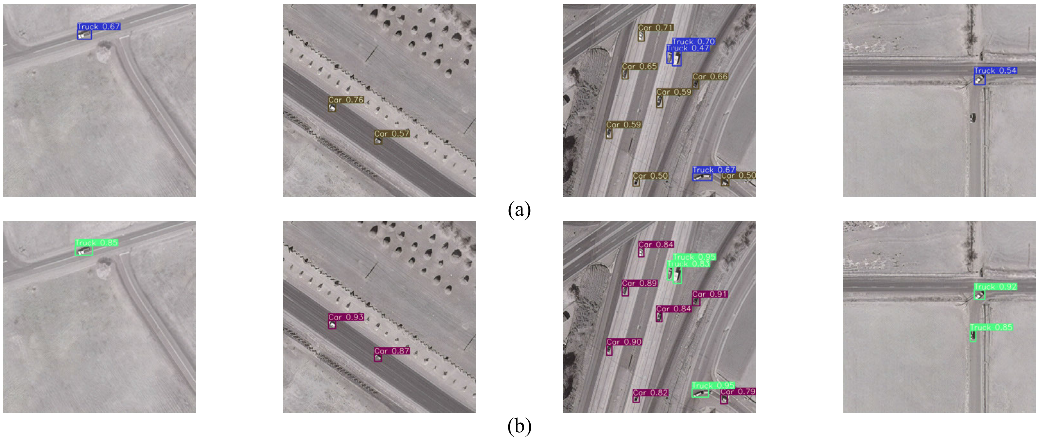
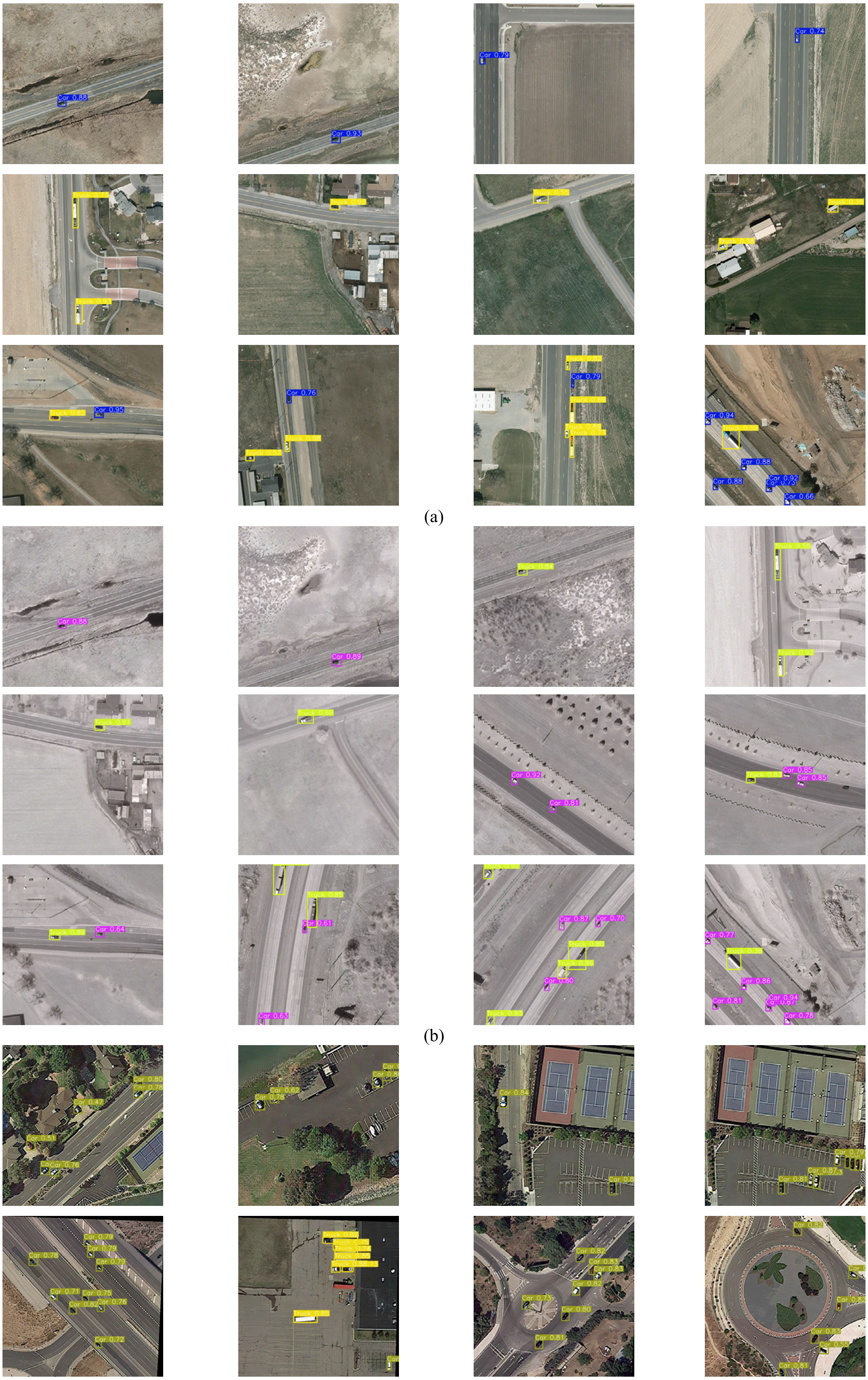


FIGURE 9. Detection results on embedded system for VEDAI-IR, (a). conventional TL, (b). Proposed method.

We compared two embedded boards: Nvidia's Jetson AGX Orin and the Jetson Nano. The average inference times for Nano and AGX Orin were 0.7867, 0.07879, 0.7847, 0.07276, 1.3458, and 0.07585, and the average NMS times were



(a)

(b)

(c)

FIGURE 10. Detection results on embedded system of the proposed method, (a). VEDIA-VISIBLE, (b) VEDIA-IR, and (c). DOTA.

0.0239, 0.00371, 0.0412, 0.00370, 0.2143, and 0.00391. The runtime of AGX Orin was 10 times faster than that of Nano. The tested remote-sensing images are shown in Figs 7, 8, 9, and 10.

G. VISUALIZATION ANALYSIS ON THE EMBEDDED SYSTEMS

This study leverages pre-trained weights of the model using the proposed method on DOTA, VEDAI-VISIBLE, and VEDAI-IR datasets with remote-sensing images. The results of the object detection accuracy of the embedded system are shown in Figs. 7, 8, 9, and 10, with Figs. 7(a), (c), 8(a), and 9(a), showcasing the outcomes of the conventional TL. In contrast, Figs. 7(b), (d), 8(b), 9(b), 10(a), 10(b), and 10(c) showcase the results of the proposed method.

V. CONCLUSION

This study proposed a novel approach for enhancing object detection performance. This method involved training the model by sequentially transferring knowledge, allowing it to learn progressively from the simplest to the most complex data. Furthermore, we enhanced object detection by adapting a pre-trained model for image generation using a GAN. The proposed method learned more robust features, enhancing generalizability and convergence. Experimental results using benchmark datasets demonstrated the effectiveness of the proposed approach, showcasing higher accuracy in object detection and localization, particularly in challenging scenarios involving LR images. In addition, the trained model of the proposed method was deployed on embedded platforms, such as Nvidia's Jetson Nano and Jetson AGX Orin for real-time remote-sensing object detection. Finally, experimental results revealed that the proposed method outperformed conventional methods in terms of accuracy.

REFERENCES

[1] X. Wu, D. Sahoo, and S. C. H. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, Jul. 2020, doi: 10.1016/j.neucom.2020.01.085.

[2] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58443–58469, 2020, doi: 10.1109/ACCESS.2020.2983149.

[3] S. Akcay, M. E. Kundegorski, C. G. Willcocks, and T. P. Breckon, "Using deep convolutional neural network architectures for object classification and detection within X-ray baggage security imagery," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 9, pp. 2203–2215, Sep. 2018, doi: 10.1109/TIFS.2018.2812196.

[4] N. Corby, "Machine vision for robotics," *IEEE Trans. Ind. Electron.*, vols. IE-30, no. 3, pp. 1–16, Aug. 1983, doi: 10.1109/TIE.1983.-356739.

[5] C. Jin, J. K. Udupa, L. Zhao, Y. Tong, D. Odhner, G. Pednekar, S. Nag, S. Lewis, N. Poole, S. Mannikeri, S. Govindasamy, A. Singh, J. Camaratta, S. Owens, and D. A. Torigian, "Object recognition in medical images via anatomy-guided deep learning," *Med. Image Anal.*, vol. 81, Oct. 2022, Art. no. 102527, doi: 10.1016/j.media.2022.102527.

[6] K. Liu and G. Mattyus, "Fast multiclass vehicle detection on aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 9, pp. 1938–1942, Sep. 2015, doi: 10.1109/LGRS.2015.2439517.

[7] H. Yan, "Aircraft detection in remote sensing images using centre-based proposal regions and invariant features," *Remote Sens. Lett.*, vol. 11, no. 8, pp. 787–796, Jun. 2020, doi: 10.1080/2150704x.2020.1770364.

[8] W. Jiang, L. Zhao, Y.-J. Wang, W. Liu, and B.-D. Liu, "U-shaped attention connection network for remote-sensing image super-resolution," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, Nov. 2022, doi: 10.1109/LGRS.2021.3127988.

[9] S. Anwar, S. Khan, and N. Barnes, "A deep journey into super-resolution: A survey," *ACM Comput. Surveys*, vol. 53, no. 3, pp. 1–34, May 2020, doi: 10.1145/3390462.

[10] Y. R. Musunuri and O. Kwon, "Super-resolution using deep residual network with spectral normalization," *Electron. Lett.*, vol. 59, no. 3, pp. 1–3, Feb. 2023, doi: 10.1049/ell2.12734.

[11] Y. Musunuri, H. Bin, and O. Kwon, "Small object detection based on YOLOv5 and super-resolution on aerial images," in *Proc. 35th Workshop Image Process. Image Understand.*, vol. 2023, pp. 1–2.

[12] W. Zhang, R. Li, T. Zeng, Q. Sun, S. Kumar, J. Ye, and S. Ji, "Deep model based transfer and multi-task learning for biological image analysis," *IEEE Trans. Big Data*, vol. 6, no. 2, pp. 322–333, Jun. 2020, doi: 10.1109/TBDATA.2016.2573280.

[13] S. Niu, Y. Liu, J. Wang, and H. Song, "A decade survey of transfer learning (2010–2020)," *IEEE Trans. Artif. Intell.*, vol. 1, no. 2, pp. 151–166, Oct. 2020, doi: 10.1109/TAI.2021.3054609.

[14] M. Mostofa, S. N. Ferdous, B. S. Riggan, and N. M. Nasrabadi, "Joint-SRVDNet: Joint super resolution and vehicle detection network," *IEEE Access*, vol. 8, pp. 82306–82319, 2020, doi: 10.1109/ACCESS.2020.2990870.

[15] Y. Musunuri, O. Kwon, and S. Y. Kung, "SRODNet: Object detection network based on super resolution for autonomous vehicles," *Remote Sens.*, vol. 14, no. 24, pp. 1–19, Dec. 2022.

[16] J. Lian, Y. Yin, L. Li, Z. Wang, and Y. Zhou, "Small object detection in traffic scenes based on attention feature fusion," *Sensors*, vol. 21, no. 9, pp. 1–16, 3390.

[17] J. Mu, S. Li, Z. Liu, and Y. Zhou, "Integration of gradient guidance and edge enhancement into super-resolution for small object detection in aerial images," *IET Image Process.*, vol. 15, no. 13, pp. 3037–3052, Nov. 2021, doi: 10.1049/ipr2.12288.

[18] Z. Yan, X. Song, H. Zhong, and X. Zhu, "Object detection in optical remote sensing images based on transfer learning convolutional neural networks," in *Proc. 5th IEEE Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, Nov. 2018, pp. 935–942.

[19] A. M. Hilal, F. N. Al-Wesabi, K. J. Alzahrani, M. A. Duhayyim, M. A. Hamza, M. Rizwanullah, and V. G. Díaz, "Deep transfer learning based fusion model for environmental remote sensing image classification model," *Eur. J. Remote Sens.*, vol. 55, no. sup1, pp. 12–23, Jan. 2022.

[20] X. Xu, H. Zhang, Y. Ma, K. Liu, H. Bao, and X. Qian, "TranSDet: Toward effective transfer learning for small-object detection," *Remote Sens.*, vol. 15, no. 14, p. 3525, Jul. 2023.

[21] K. Suseela and K. Kalimuthu, "An efficient transfer learning-based super-resolution model for medical ultrasound image," *J. Phys., Conf.*, vol. 1, pp. 1–8, Sep. 2021, doi: 10.1088/1742-6596/1964/6/062050.

[22] Y. Huang, Z. Jiang, R. Lan, S. Zhang, and K. Pi, "Infrared image super-resolution via transfer learning and PSRGAN," *IEEE Signal Process. Lett.*, vol. 28, pp. 982–986, 2021, doi: 10.1109/LSP.2021.3077801.

[23] J. Talukdar, S. Gupta, P. S. Rajpura, and R. S. Hegde, "Transfer learning for object detection using state-of-the-art deep neural networks," in *Proc. 5th Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Feb. 2018, pp. 78–83, doi: 10.1109/SPIN.2018.8474198.

[24] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1–11.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg, "SSD: Single shot multi box detector," in *Proc. Eur. Conf. Comput. Vis. ECCV*, 2016, pp. 1–17.

[26] J. Li, H.-C. Wong, S.-L. Lo, and Y. Xin, "Multiple object detection by a deformable part-based model and an R-CNN," *IEEE Signal Process. Lett.*, vol. 25, no. 2, pp. 288–292, Feb. 2018, doi: 10.1109/LSP.2017.2789325.

[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," 2015, *arXiv:1506.01497*.

[28] L. Courtrai, M.-T. Pham, and S. Lefèvre, "Small object detection in remote sensing images based on super-resolution with auxiliary generative adversarial networks," *Remote Sens.*, vol. 12, no. 19, p. 3152, Sep. 2020.

[29] L. Chen, W. Shi, and D. Deng, "Improved YOLOv3 based on attention mechanism for fast and accurate ship detection in optical remote sensing images," *Remote Sens.*, vol. 13, no. 4, p. 660, Feb. 2021.

[30] W. Wang, X. Tan, P. Zhang, and X. Wang, "A CBAM based multiscale transformer fusion approach for remote sensing image change detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 6817–6825, 2022.

[31] Y. Gong, Z. Xiao, X. Tan, H. Sui, C. Xu, H. Duan, and D. Li, "Context-aware convolutional neural network for object detection in VHR remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 34–44, Jan. 2020, doi: 10.1109/TGRS.2019.2930246.

[32] Y. Liu, J. Yao, X. Lu, M. Xia, X. Wang, and Y. Liu, "RoadNet: Learning to comprehensively analyze road networks in complex urban scenes from high-resolution remotely sensed images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2043–2056, Apr. 2019, doi: 10.1109/TGRS.2018.2870871.

[33] H. Tayara, K. Gil Soo, and K. T. Chong, "Vehicle detection and counting in high-resolution aerial images using convolutional regression neural network," *IEEE Access*, vol. 6, pp. 2220–2230, 2018, doi: 10.1109/ACCESS.2017.2782260.

[34] J. Rabbi, N. Ray, M. Schubert, S. Chowdhury, and D. Chao, "Small-object detection in remote sensing images with end-to-end edge-enhanced GAN and object detector network," *Remote Sens.*, vol. 12, no. 9, p. 1432, May 2020, doi: 10.3390/rs12091432.

[35] Z. Dong, M. Wang, Y. Wang, Y. Zhu, and Z. Zhang, "Object detection in high resolution remote sensing imagery based on convolutional neural networks with suitable object scale features," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 3, pp. 2104–2114, Mar. 2020, doi: 10.1109/TGRS.2019.2953119.

[36] M. Rifat Arefin, V. Michalski, P.-L. St-Charles, A. Kalaitzis, S. Kim, S. E. Kahou, and Y. Bengio, "Multi-image super-resolution for remote sensing using deep recurrent networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 816–825, doi: 10.1109/CVPRW50498.2020.00111.

[37] R. Fernandez-Beltran, P. Latorre-Carmona, and F. Pla, "Single-frame super-resolution in remote sensing: A practical overview," *Int. J. Remote Sens.*, vol. 38, no. 1, pp. 314–354, Jan. 2017, doi: 10.1080/01431161.2016.1264027.

[38] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," 2018, *arXiv:1808.01974*.

[39] B. Pan, J. Tai, Q. Zheng, and S. Zhao, "Cascade convolutional neural network based on transfer-learning for aircraft detection on high-resolution remote sensing images," *J. Sensors*, vol. 2017, pp. 1–14, Jul. 2017, doi: 10.1155/2017/1796728.

[40] J. Chen, X. Liu, C. Liu, Y. Yang, S. Yang, and Z. Zhang, "A modified convolutional neural network with transfer learning for road extraction from remote sensing imagery," in *Proc. Chin. Autom. Congr.*, 2018, Nov. 2018, pp. 4263–4267, doi: 10.1109/CAC.2018.8623081.

[41] X. Tong, X. Xu, S. Huang, and L. Zheng, "A Mathematical frame work for quantifying transferability in multi-source transfer learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 1–14.

[42] W. Liu, K. Quijano, and M. M. Crawford, "YOLOv5-Tassel: Detecting tassels in RGB UAV imagery with improved YOLOv5 based on transfer learning," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 8085–8094, 2022.

[43] X. Li, Z. Wang, S. Geng, L. Wang, H. Zhang, L. Liu, and D. Li, "YOLOv3-Pruning(transfer): Real-time object detection algorithm based on transfer learning," *J. Real-Time Image Process.*, vol. 19, no. 4, pp. 839–852, Jun. 2022.

[44] D. Hong, B. Zhang, X. Li, Y. Li, C. Li, J. Yao, N. Yokoya, H. Li, P. Ghamisi, X. Jia, A. Plaza, P. Gamba, J. A. Benediktsson, and J. Chanussot, "Spectral GPT: Spectral Remote Sensing Foundation Model," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 1, 2024, doi: 10.1109/TPAMI.2024.3362475.

[45] C. Li, B. Zhang, D. Hong, J. Yao, and J. Chanussot, "LRR-Net: An interpretable deep unfolding network for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5, doi: 10.1109/TGRS.2023.3279834.

[46] X. Wu, D. Hong, and J. Chanussot, "UIU-Net: U-Net in U-Net for infrared small object detection," *IEEE Trans. Image Process.*, vol. 32, pp. 364–376, 2023, doi: 10.1109/TIP.2022.3228497.

[47] X. Wu, D. Hong, J. Tian, J. Chanussot, W. Li, and R. Tao, "ORSIm detector: A novel object detection framework in optical remote sensing imagery using spatial-frequency channel features," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 5146–5158, Jul. 2019, doi: 10.1109/TGRS.2019.2897139.

[48] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2005, *arXiv.2005.12872*.

[49] C. Feng, Y. Zhong, Y. Gao, M. R. Scott, and W. Huang, "TOOD: Task-aligned one-stage object detection," 2021, *arXiv:2108.07755*.

[50] V. Mazzia, A. Khaliq, F. Salvetti, and M. Chiaberge, "Real-time apple detection system using embedded systems with hardware accelerators: An edge AI application," *IEEE Access*, vol. 8, pp. 9102–9114, 2020.

[51] D. Saggu and A. Azim, "Transfer learning on the edge networks," in *Proc. IEEE Int. Syst. Conf.*, Apr. 2021, pp. 1–8.

[52] H. Choi, Y. Musunuri, and O. Kwon, "Object recognition algorithm based on complex transfer learning on remote sensing images," in *Proc. Korea Multimedia Soc. Fall Conf.*, 2023, pp. 1–2.

[53] Y. Musunuri, H. Choi, and O. Kwon, "Target detection method based on embedded systems for aerial images," in *Proc. 36th Workshop Image Process. Image Understand.*, 2024, pp. 1–2.

[54] I. Koshelev, M. Savinov, A. Menshchikov, and A. Somov, "Drone-aided detection of weeds: Transfer learning for embedded image processing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 102–111, 2023.

[55] I. Athanasiadis, P. Mousouliotis, and L. Petrou, "A framework of transfer learning in object detection for embedded systems," 2018, *arXiv:1811.04863*.

[56] K. K. Gadiraju and R. R. Vatsavai, "Remote sensing based crop type classification via deep transfer learning," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 4699–4712, 2023.

[57] X. Ma, K. Ji, B. Xiong, L. Zhang, S. Feng, and G. Kuang, "Light-YOLOv4: An edge-device oriented target detection method for remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 10808–10820, 2021, doi: 10.1109/JSTARS.2021.3120009.

[58] Z. Sun, X. Leng, Y. Lei, B. Xiong, K. Ji, and G. Kuang, "BiFA-YOLO: A novel YOLO-based method for arbitrary-oriented ship detection in high-resolution SAR images," *Remote Sens.*, vol. 13, no. 21, p. 4209, Oct. 2021.

[59] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. Loy, Y. Qiao, and X. Tang, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 1–13.

[60] C.-Y. Wang, A. Bochkovskiy, and H.-Y. Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, *arXiv:2207.02696*.

[61] Y. Wu, A. Kirillov, F. Massa, W. Lo, R. Girshick. (2019). *Detectron2*. [Online]. Available: https://github.com/facebookresearch/detectron2

[62] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2017, *arXiv:1708.02002*.

[63] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with YOLOv8," 2023, *arXiv:2305.09972*.

YOGENDRA RAO MUSUNURI received the B.Tech. degree in electronics and communication engineering from DPREC, Jawaharlal Nehru Technological University (JNTU), Hyderabad, India, in 2012, and the M.E. degree in image processing and computer vision from Busan University of Foreign Studies (BUFS), Busan, South Korea, in 2017. He is currently pursuing the Ph.D. degree in the control and instrumentation engineering with Changwon National University (CWNU), Changwon, South Korea. His research interests include image processing, computer vision, object detection, super-resolution (SR), advanced driving assistant systems (ADAS), remote sensing, target detection, XAI, edge AI, machine learning, and deep learning.

CHANGWON KIM received the B.S., M.S., and Ph.D. degrees in the electrical and electronic engineering from Yonsei University, Republic of Korea, in 2002, 2004, and 2010, respectively. From 2010 to 2015, he was a Senior Researcher with Samsung Electronics, Suwon, South Korea. From 2015 to 2022, he was a Patent Examiner with Korean Intellectual Property Office. Since 2022, he has been with Changwon National University, where he is currently an Assistant Professor. His current research interests include sensor signal processing, image and video processing, and computer vision.

OH-SEOL KWON received the B.S. and M.S. degrees in electrical engineering and computer science and the Ph.D. degree in electronics from Kyungpook National University, Republic of Korea, in 2002, 2004, and 2008, respectively. From 2008 to 2010, he was a Postdoctoral Research Fellow with New York University, New York, NY, USA. From 2010 to 2011, he was a Senior Researcher with the Visual Display Division, Samsung Electronics, Suwon, South Korea. He was a Visiting Professor with Princeton University, Princeton, NJ, USA. He joined Changwon National University, in 2011, and is currently a Professor. His research interests include signal processing, remote sensing, computer vision, deep learning, and human visual systems.

SUN-YUAN KUNG (Life Fellow, IEEE) is currently a Professor with the Department of Electrical Engineering, Princeton University, Princeton, NJ, USA. He has authored or co-authored over 500 technical publications and numerous textbooks, including the *VLSI Array Processors* (Prentice-Hall, 1988), the *Digital Neural Networks* (Prentice-Hall, 1993), the *Principal Component Neural Networks* (Wiley, 1996), the *Biometric Authentication: A Machine Learning Approach* (Prentice-Hall, 2004), and the *Kernel Methods and Machine Learning* (Cambridge University Press, 2014). His current research interests include machine learning, data mining and privacy, statistical estimation, system identification, wireless communication, VLSI array processors, signal processing, and multimedia information processing. He was a Founding Member of several technical committees of the IEEE Signal Processing Society. He served as a member for the Board of Governor of the IEEE Signal Processing Society, from 1989 to 1991. He was a recipient of the IEEE Signal Processing Society's Technical Achievement Award for the contributions on parallel processing and neural network algorithms for signal processing in 1992, the Distinguished Lecturer of the IEEE Signal Processing Society in 1994, the IEEE Signal Processing Society's Best Paper Award for his publication on principal component neural networks in 1996, and the IEEE Third Millennium Medal in 2000. He was an Associate Editor of VLSI and *Neural Networks* of IEEE TRANSACTIONS ON SIGNAL PROCESSING in 1984 and 1991, respectively. He was a Distinguished Lecturer of the IEEE Signal Processing Society in 1994. Since 1990, he has been the Editor-in-Chief of the *Journal of VLSI Signal Processing Systems*.