Object detection for low-resolution image using GAN-Yolo-4 for UAV aerial images data

[1]Vikul Kumar, [2]Kajal Kori

[1]Lecturer, [2]Lecturer,
[1] Department of CSE and AI
[1] SD college of Engineering and Technology, Muzaffarnagar, UP, India
[1]Vikulkumar9262@gmail.com, [2]Kajalkori091@gmail.com

***Abstract*— Aerial photography object identification by unmanned aerial vehicles (UAVs) is crucial for disaster relief, environmental protection and military monitoring. Broad photography, which adds background noise, and high imaging height, which makes it more likely that there may be little things in the images, are the problems. In order to get around this problem, remote sensing has resorted to super-resolution Generative Adversarial Networks (GANs), specifically enhanced super-resolution GANs (ESRGAN) and EESRGAN, which excel in spotting small objects in fuzzy images. Due to this accomplishment, we suggest employing comparable ESRGAN approaches to improve photos of road surfaces. We want to enhance the image quality of remote sensing images using residual-in-residual dense blocks (RRDB). We develop edge-enhanced super-resolution GAN (EESRGAN), using edge enhancement methods, to further improve the pictures and identify small objects. We use different detector networks, such as Yolo-4, SSD and FRCNN in an end-to-end method, which is essential for reliably recognizing object. The performance of detection is improved by backpropagating the detector loss into EESRGAN. We thoroughly test our methods using datasets such as COWC, CIFAR-10, and COCO.**

***Index Terms*—You Only Look Once (Yolo-4 v4),GAN, lowresolution (LR) image classification, single-shot multibox detector (SSD), faster region-based convolutional neural network (FRCNN), edge improvement, satellite imagery, and high-resolution**

I. INTRODUCTION

The ability to identify objects in remote sensing data has enormous promise in a number of fields, including traffic management, forestry, environmental control, surveillance, military, national security and monitoring oil or gas operations. For identifying and finding objects in photographs taken using satellite or drone technology, a variety of techniques have been developed. However, these techniques often fail to perform well when dealing with noise as well as low-resolution (LR) pictures, particularly while handling little items. Small item identification performance is often worse than bigger object detection even in high-resolution (HR) pictures.

Researchers have used super-resolution (SR) approaches based on deep convolutional neural networks (CNN) to improve the accuracy of object recognition in remote sensing photos. These methods use SR models to create fake pictures, which are subsequently augmented images are used to conduct object detection. Deep CNN-based SR methods, such as a single picture super-resolution convolutional networks (SRCNN) and correct picture super-resolution using very deep convolutional networks (VDSR), have shown impressive results in producing high-resolution (HR) imagery that closely resembles the ground truth from low-resolution (LR) input data.

Super-resolution GAN (SRGAN) and enhanced superresolution GAN (ESRGAN), two LR image improvement approaches based on Generative Adversarial Networks (GANs), have also shown excellent performance. These models are made up of two deep CNN-based subnetworks, a generator and a discriminator. For the training and evaluation of the algorithms, datasets containing pairs of LR as well as HR images are employed. Discriminator is taught to discriminate between authentic HR photos and upscaled LR images, and the generator seeks to produce HR images based on the LR input. Through practice, the generator develops the ability to produce HR pictures that closely mimic the actual situation, making it challenging for the discriminator to distinguish between the two.

Researchers have suggested numerous approaches employing different deep convolutional neural network (CNN) edge extractors to provide distinct and unambiguous edge information. When used on low-resolution (LR) and noisy remote sensing photos, these approaches' performance tends to suffer even when they provide acceptable results for natural images. In a recent method, an edge-enhancement network (EEGAN) based on GAN was presented to provide aesthetically appealing outcomes with enough edge information. Within its generator design, EEGAN uses two subnetworks: the first network creates intermediate high-resolution (HR) pictures, while the second network creates crisp, noise-free edges based on these intermediate images.

This method captures edge information and generates noisefree edges by combining a mask branch with Laplacian function. Although this method effectively maintains sufficient edge information, it has the disadvantage that the final output pictures could be less sharp than they would be using a modern GAN-based super-resolution (SR) technique. This blurriness is a result of noise being added at the improved borders, which can affect how well object identification works.

In several computer vision applications, such as UAVbased video surveillance, remote sensing for Earth vision, and privacy-preserving video analysis, the identification of objects in low-resolution (LR) pictures is of vital relevance. Numerous CNN-based object identification algorithms have been developed to extract discriminative features from regions of interest (RoIs) and categorize photos as a result of convolutional neural network (CNN) improvements.These methods work well on high-resolution (HR) photos with plenty of details, but when used on very low-resolution (LR) images, their performance noticeably suffers. **Type Style and Fonts**

Wherever Times is specified, Times Roman or Times New Roman may be used. If neither is available on your word processor, please use the font closest in appearance to Times. Avoid using bit-mapped fonts. True Type 1 or Open Type fonts are required. Please embed all fonts, in particular symbol fonts, as well, for math, etc.

II. RELATED WORKSE

Our study uses an integrated system made up of an object detection network and an edge-enhanced image superresolution (SR) network. We shall examine the current approaches that are pertinent to our work in this part.

A. Object Detection

Unified models, which use single-stage detection methods and region-based CNN (R-CNN) models, which employ a twophase identification process, are the two subclasses of deep learning-based object detectors [1]. R-CNN, Fast R-CNN, and Faster R-CNN are examples of two-phase detectors, whereas SSD, You Only Look Once (Yolo-4) and RetinaNet are wellknown single-stage detectors [1]. In the case of two-stage detectors, the first step entails choosing areas of interest using a region proposal network or a selective search. In the second step, these chosen locations are then checked for certain item kinds, and minimum bounding boundaries are anticipated for the objects found. Instead, single-stage detectors execute detection on a dense sample of all potential sites without the requirement for a region proposal network. Single-stage detectors are thus quicker but often less precise than their twostage equivalents. However, A focus loss algorithm is used by RetinaNet to address the issue of data imbalance caused by numerous backdrop objects. This method frequently operates in a manner similar to two-stage techniques [2].

The researchers unveiled an end-to-end aircraft detector that uses transfer learning and is based on the Single Shot MultiBox Detector (SSD) architecture. They suggested a technique involving the breaking up of huge photos into smaller tiles in order to get around the restrictions brought on by the restricted availability of a limited number of aircraft images for training. Following the discovery of objects on these smaller tiles, each picture tile was mapped back to the original image. When compared to the conventional SSD model, their method performed better [3]. In a different research [4], the authors concentrated on improving convolutional neural networks' (CNNs') object identification capabilities particularly for remote sensing pictures. They found that attaining better outcomes depended greatly on adjusting parameter values. They adjusted the settings using the You Only Look Once (Yolo-4) model as the object detector to improve the results of inference and detection [4].

Yolo-4 immediately predicts object bounding boxes and class labels without the need for extra processing stages, in contrast to two-stage detectors that depend on region proposal networks. As a result, Yolo-4 has a shorter inference time than its competitors. The precision of recognizing tiny things or objects in busy situations may be compromised, however. Yolo-4 has gone through a number of revisions, with each one seeking to enhance detection performance and fix the shortcomings of the one before it. Due to its real-time capabilities and capacity to identify several things at once, it has found use in a number of fields, including autonomous driving, surveillance, and robots.

B. Image-Enhancement-Based Techniques Super-Resolution of Images

Several super-resolution (SR) methods have been developed using deep convolutional neural networks (CNNs). By using end-to-end training, Dong et al.'s super-resolution CNN (SRCNN) [5] was able to significantly improve low-resolution (LR) pictures. Deep CNN architectures for SR advanced quickly as the area developed. Researchers created methods like residual blocks [6], densely connected networks [7], and leftover dense blocks [8] to further improve SR outcomes.Deep CNNs without the batch normalization (BN) layer were used in Huang et al.'s [9] and Lim et al.'s [10] trials, which produced significant performance benefits and reliable training, particularly in deeper networks. These advancements mostly focused on natural photos, showing how deep CNNs may improve the visual quality and details of LR photographs.

Reconstructing high-quality pictures from their lowresolution (LR) counterparts is the goal of imageenhancement-based approaches, which are designed to boost recognition performance. Image super-resolution (SR) modules have been used in earlier works like [11], [12], and [13] to convert low-resolution (LR) pictures into visually realistic high-resolution (HR) images for future classification tasks. The SR module and classification module are often tuned separately, which makes it difficult for these approaches to provide an optimum solution for the classification problem. A multitask generative adversarial network (GAN)-based system was presented in [14] in an attempt to solve this shortcoming. This system used a generator model based on SR to transform LR regions of interest (RoIs) into highquality representations, enhancing the detailed information. Additionally, bounding box classification and regression tasks were carried out using a multitask learning model as the discriminator. This method sought to improve the overall performance of object detection and identification tasks by merging the SR and classification goals inside a cohesive framework.

C. Combining Object Detection and Super-Resolution

The literature has looked at the possible advantages of super-resolution (SR) in object detection tasks. The authors of [15] performed tests utilizing remote sensing datasets to look at how SR improved item recognition. Another work [16] coupled object detection with single-shot multibox detector (SSD) and Yolo-4 object augmentation utilizing CNN-based picture enhancement. A deep CNN-based generator that converts low-resolution (LR) pictures into high-resolution (HR) images was presented by Haris et al. [17]. They also suggested a multi-task network that can perform object localisation and categorization in addition to acting as a discriminator. These techniques required training on pairs of LR and HR photos, and they were largely used on natural images. A simultaneous super-resolution and object recognition technique was put out particularly for satellite pictures in a separate strategy [13]. This method used a redesigned, quicker R-CNN architecture for vehicle recognition together with an SR network influenced by the cycle-consistent adversarial network. With the use of the SR network and object identification framework, it was intended to improve satellite photos. These studies demonstrate the possibility of integrating SR with object identification methods to improve item recognition quality and precision in both natural and satellite imaging domains.

C. Methods Based on Representation Enhancement

By including procedures that are either resolution-invariant or change the representations, representation-enhancement based approaches seek to improve the discriminative capacities of representations taken from low-resolution (LR) pictures. These methods concentrate on bridging the gap between representations with high resolution (HR) and those with low resolution (LR). One class of techniques, known as representation-translating techniques, reduces the separation between the HR and LR representations by transforming them into a single shared representation space [18]–[19]. In order to overcome the difficulties of extremely high LR

recognition, CNN models were used in [20], which represents the field's groundbreaking achievement. A sparse architecture is used by Wei et al. [21] in a different work to propose a sparse image transformation technique that captures the connection between LR pictures and their associated HR equivalents. By either making the representations resolution-invariant or altering them to better match their HR counterparts, these representation-enhancement-based techniques seek to boost the recognition performance on LR pictures. They do this to improve the representations retrieved from LR pictures' ability to discriminate.

D. Deep Learning's Use of Spectral Bias

The F-principle was suggested by Xu et al. in their article [22], which also claimed that deep neural networks often train using a mapping function that favors low to high frequencies. The response frequency, which quantifies the rate of change of the mapping function depending on the intensity of each pixel, is referred to as "frequency" in this context. Using Fourier analysis, Xu et al. [23] expanded the theoretical framework and showed that convolutional neural networks (CNNs) give low-resolution (LR) components priority during training. They discovered that for datasets taken from the actual world, CNNs initially catch the dominating low-frequency (LF) components before slowly but steadily capturing the high-frequency (HF) components. This finding points to the spectral bias, a learning bias in deep neural networks that Rahaman et al. [24] have emphasized. template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

III. METHODOLOGY

The generator (G), the discriminator (DRa), and the edgeenhancement network (EESN) are the three parts that make up the SR network in our approach. Our end-to-end training methodology includes backpropagating into a generator the variety of recognizing the loss. This implies that the detector serves as a discriminator as well, directing the generator G to produce realistic pictures that closely reflect the actual world. The discriminator, which consists of the DRa and the detector network, and the generator, which is made up of the EEN, are the two main components of the overall network structure. The detector's function as a discriminator in our method is shown in Figure 1.

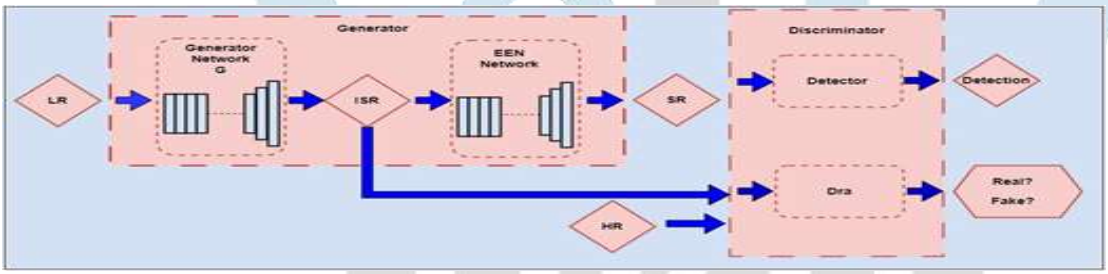


Fig. 1. Network design as a whole includes a generator and a discriminator module.

A. Generator Network G

- We use the RRDB (Residual-in-Residual Dense Blocks) generator design from ESRGAN [25], which eliminates all batch normalization (BN) stages. Figure 2 shows the general layout of our generator G, while Figure 3 and 4 shows the RRDB architecture.

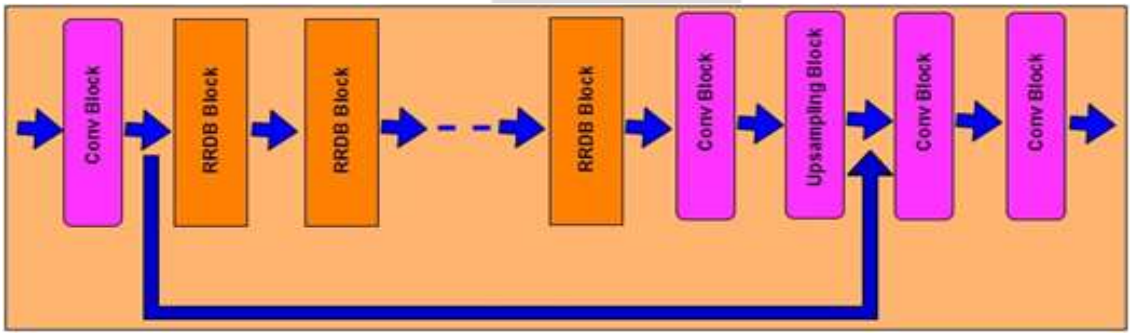


Fig. 2. Convolutional, upsampling, and residual-in-residual dense blocks in Generator G

Inspire by ESRGAN, we remove BN layers to improve generator G's performance and simplify calculation. BN layers may create undesired artifacts and impair the generator's capacity to generalize, according to the developers of ESRGAN, when there are large statistical differences between the training and testing datasets.

We construct a generator network G utilizing RRDB as the main component. The capacity of the network is efficiently increased by these blocks, which create a multi-level

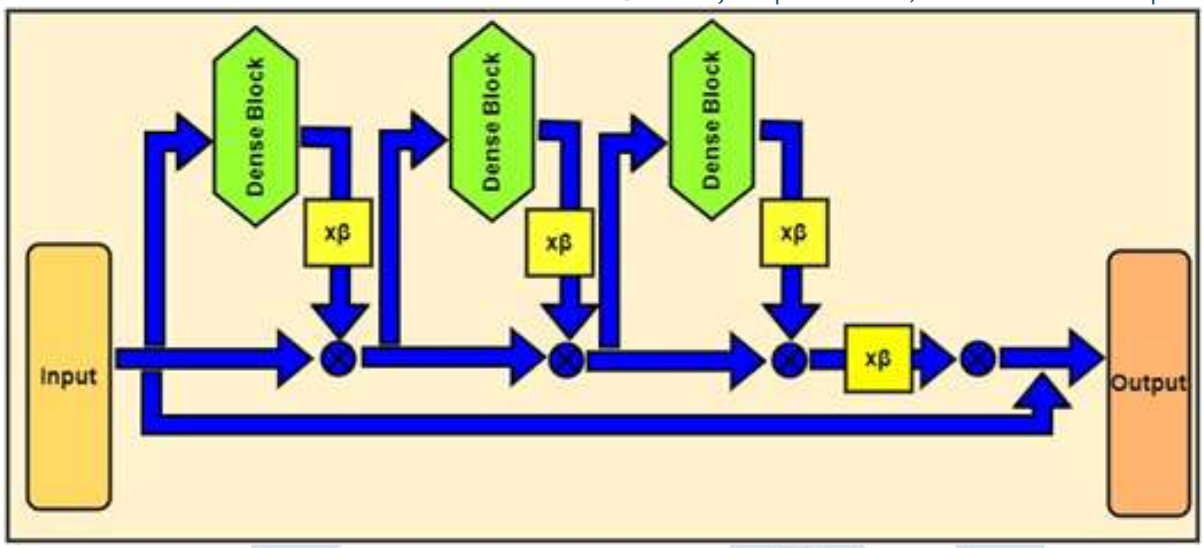


Fig. 3. RRDB from generator.

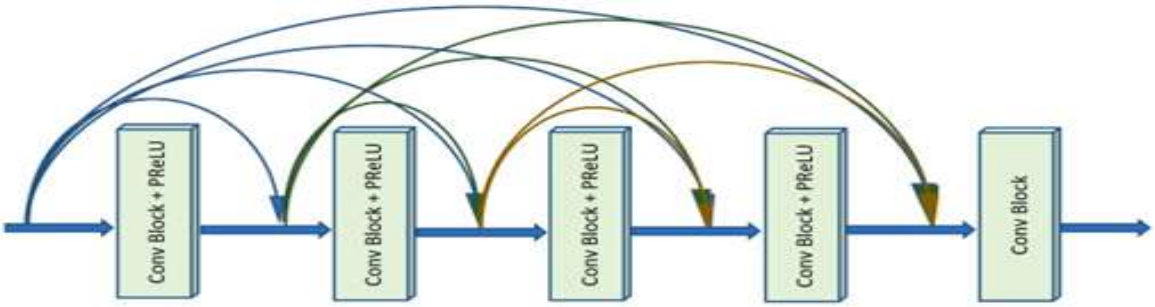


Fig. 4. Dense block from RRDB..

residual network with many dense connections. To provide steady circumstances throughout the training phase, we also use residual scaling [25].We train the parameters for the dense blocks in RRDB using the parameterized rectification linearity unit (PReLU) [23] in conjunction with other neural network parameters. We use relativity median discrimination for the discriminator DRa, which is equivalent to the approach outlined in [25].

Equations (1) and (2) provide the relativistic average discriminator formulation for our design. Before presenting further information in Section C, we will quickly touch on the discriminator DRa, which the generator G depends on. The discriminator's job is to determine whether a genuine image

IHR or an artificially created intermediate image IISR is more realistic.

DRa (I HR,IISR) = σ (C (IHR) − EIISR [C (IISR)]) 1 More Realistic than fake data? (1) →

DRa (IISR,IHR = σ (C (IISR) − EIHR [C (IHR)])

0 Less realistic than real data? (2) →

In equations (1) and (2), the symbols σ,C(·) and EIISR denote the sigmoid function, discriminator output, and the calculation of the mean for all intermediate images generated within a mini-batch, respectively.The generator, where IISR = G(ILR), creates these intermediate pictures. Equation

(3) shows that the generator may use gradients from both the produced and ground truth pictures during training since its adversarial loss includes both IHR and IISR. Equation (4) provides an example of the discriminator's loss function.

LRaG = −EIHR [log(1 − DRa (IHR,IISR))]−

(3) EIISR [log(DRa (IISR,IHR))]

LRaD = −EIHR [log(DRa (IHR,IISR))]−

(4)

EIISR [log(1 − DRa (IISR,IHR))]

We use the perceptual loss (Lpercep) and the content loss (L1) as additional loss functions to improve the generator G's performance [21]. The feature map (vggfea(·)) derived from a fine-tuned VGG19 [26] network before the activation layers is evaluated to determine the perceptual loss. The content loss, on the other hand, gauges the deviation from the 1-norm between

IISR and IHR. Equations (5) and (6), respectively, reveal the formulations of the perceptual loss and the content loss.

Lpercep = EILR // vggfea (G(ILR) − vggfea (IHR) // 1(5)

L1 = EILR // G(ILR) − IHR // 1 (6)

B. Edge-Enhancement Network EESRGAN

The EEN (Edge Enhancement Network) is designed to take away noise and improve the image's extracted edges. The Laplacian operator [28] is used first to obtain the boundary lines from the input image. The edge data is first acquired, and then it is processed further using convolutional procedures, RRDB (Residual-in-Residual Dense Blocks) and extra sampling blocks. Edge noise is reduced using a masking node with quadratic modulation as discussed in [22]. Once the lines left behind by the Laplacian operator have been removed, the enhanced edges are then reintegrated with the original pictures.

The edge-enhancement subnetwork proposed in [22] differs from the EEN network in two ways. First, we replace the large number of blocks with RRDB, which according to ESRGAN [25] provides greater performance. The performance of the EEN network will be enhanced by this replacement. Second, to enhance the rebuilt cutting-edge knowledge and further boost the final results, we include a new loss term. In a prior study [26], the authors used an edge-enhancement subnetwork to improve the edges by extracting edge information from the produced intermediate image IISR. The original edges were then subtracted, and the improved edges were then readded to the IISR. They developed a consistency loss for pictures (Limg cst) utilizing the Charbonnier loss [27], which encourages aesthetically appealing outputs with precise edge information, to train the network. However, sometimes, certain objects' edges were deformed and generated noise, leading to inadequate edge information. We also suggest a consistency loss (Ledge cst) for the edges to alleviate this issue. Ledge cst is calculated by comparing the Charbonnier loss between the edges that were retrieved from the high-resolution image IHR and the super-resolved image ISR, respectively.

To compute Ledge cst, we evaluate the Charbonnier loss between the extracted edges (Iedge SR) from ISR and the extracted edges (Iedge HR) from IHR. The consistency losses are represented by Equations (7) and (8), where the function r() represents the Charbonnier penalty function [27]. By summing the individual losses for both the edges and the images, we obtain the total consistency loss. The loss function for our EEN can be observed in Equation (9).

Limg cst = EISR [ρ(IHR − ISR)] (7)

Ledge cst = EIedge SR [ρ(Iedge HR − Iedge SR)] (8)

Leen = Limg cst + Ledge cst (9)

Combining the separate losses of the EEN network and generator G yields the generator module's overall loss. The total loss is calculated using the following equation:

Loss generator = l1*Loss adversarial + l2*Loss perceptual + l3*Loss content + l4*Loss edge(10)

Within this equation: Adversarial loss is shown as loss adversarial. Loss perceptual refers to the perceived loss. The loss of material is indicated by Loss Content. Edge loss is indicated by loss edge. These weight parameters, λ1,λ2,λ3, and λ4, are utilized to balance the various elements of the loss. Empirically, λ1 = 1,λ2 = 0.001,λ3 = 0.01, and λ4 = 5 are chosen as the values.

These discrete loses and their corresponding weight factors are added to determine the generator module's overall damage, enabling effective model training and optimization.

LG−een = λ1Lpercep + λ2LRaG + λ3L1 + λ4Leen (11)

C. Discriminator

Our model's discriminator architecture is built on ESRGAN [21], which makes use of the VGG-19 [28] architecture. For our detector networks, we use the quicker R-CNN [8], SSD [29], and Yolo-4 algorithms. The detector network and the discriminator (DRa) together serve as a discriminator for the generating module.

Two networks make up the object detection model known as the Faster R-CNN [8]. The region proposal network (RPN), the initial network, creates region suggestions from the input picture. The task of identifying items inside these suggestions and precisely fitting bounding boxes around them falls on the second network.A classifier and a regressor are the next two branches in the detection process after the region proposal network (RPN). Each suggestion is assigned to a particular object class by the classifier branch, while the regressor branch seeks to identify the exact bounding box coordinates of the item. In our example, only items from one class are present in both databases. As a consequence, just the background class and the object class are predicted by our classifier.

An effective object detection model that identifies things in a single trip over the network is the SSD (Single Shot Multibox Detector) [29].The Yolo-4 (You Only Look Once) object detection method seeks to achieve high accuracy, real-time object recognition. Yolo-4 conducts object classification and bounding box regression in a single pass through the network, in contrast to conventional object detection techniques that depend on region proposal networks. The term "single-stage" in this instance refers to the fact that categorization and localisation are carried out during the same forward pass. SSD and Yolo-4 use feature extraction networks in a manner similar to Faster R-CNN, and several kinds of networks may be used. For speed reasons, we employ the VGG-16 [26] network as our characteristic extraction in SSD and Yolo-4. The convolutional feature layers of SSD and Yolo-4, which may be thought of as a pyramid representation of pictures at various scales, are built upon this network. The object identification procedure is then completed at each layer, producing class predictions and bounding box coordinates as the end result.

1) Loss of the Discriminator: To achieve the overall discriminator loss, the relativistic discriminator loss LRda and detector loss are combined. Similar regression and localization losses are used by faster R-CNN, SSD and Yolo-4, but classification losses vary. In this localization and regression, the smooth L1 loss [8] is employed to compute the disparity between the predicted bounding box coordinates and the corresponding ground truth coordinates (t*).The classification loss

Lcls frcnn, regression loss Lreg frcnn, and overall loss Ldet frcnn of Faster R-CNN are provided below:

Lcls frcnn = X[−log(Dcls frcnn(GG een(ILR)))] (12)

ILR

Lreg frcnn = X[smoothL1(Dreg frcnn(GG een(ILR)),tc)] (13)

ILR

Ldet frcnn = Lcls frcnn + λLreg frcnn (14)

Based on actual facts, the parameter λ is adjusted to 1 in this instance to balance the losses. The terms "Detcls frcnn" and "Detreg frcnn" relate to the regression model and classification parts of Faster R-CNN, respectively. The identification loss for SSD and Yolo-4 Lcls ssd/Yolo-4, regression loss Lreg ssd/Yolo-4, and overall loss Ldet ssd/Yolo-4 are shown below:

Lcls ssd/Yolo-4 = X[−log(softmaxDcls frcnn(GG een(ILR)))]

ILR

(15)

Lreg ssd/Yolo-4 = X[smoothL1(Dreg ssd/Yolo-4(GG een(ILR)),tc)]

ILR

(16)

Ldet ssd/Yolo-4 = Lcls ssd/Yolo-4 + βLreg ssd/Yolo-4 (17)

In this instance, the loss balancing parameter β is used, and its value is experimentally set to 1. The classifier and regressor components of the SSD and Yolo-4 are denoted by the variables Detcls ssd/Yolo-4 and Detreg ssd/Yolo-4 respectively. 2) Training: The detector and SR network are trained, which consists of the generator module and the discriminator DRa, individually. In this

setup, the detector's losses is not returned to the generating unit. As a result, the generator only hears feedback from the discriminator DRa, and is not aware of the detector. Equation (11), illustrates that no error is back transmitted to the GG−een network when loss Lcls−frcnn is computed and how the network is disconnected during the calculation of the detection losses.

End-to-end training involves concurrently training the generating module and detector as well as the complete architecture. This indicates that the generator module receives a backpropagation of the detector loss. As a consequence, the discriminator DRa and the detector both send gradients to the generating module. It is possible to represent the last discriminator failure LD−det follows:

$$LD-det = LRaD + h \cdot LDdet \quad (18)$$

The detector loss contribution is balanced by the parameter h, which is experimentally adjusted at 1. Ldet is a representation of the detection loss from SSD, Yolo-4 or Faster RCNN. Finally, given our design, we derive the total loss Loverall, which can be expressed as follows:

$$Loverall = LG-een + LD-det \quad (19)$$

Two potential methods may be used to increase lowresolution picture categorization in UAV systems and identify objects in images from remote sensing.There are given belowIn the first technique, an image is captured for detection using a discriminator network, producing low-resolution (LR) pictures as a consequence. A generator network is used to convert these LR photos into SR pictures, producing intermediate super-resolution images ISR. Finally, the ESRGAN/EESRGAN network is used to improve the ISR images.Fig.1 show network's architecture:

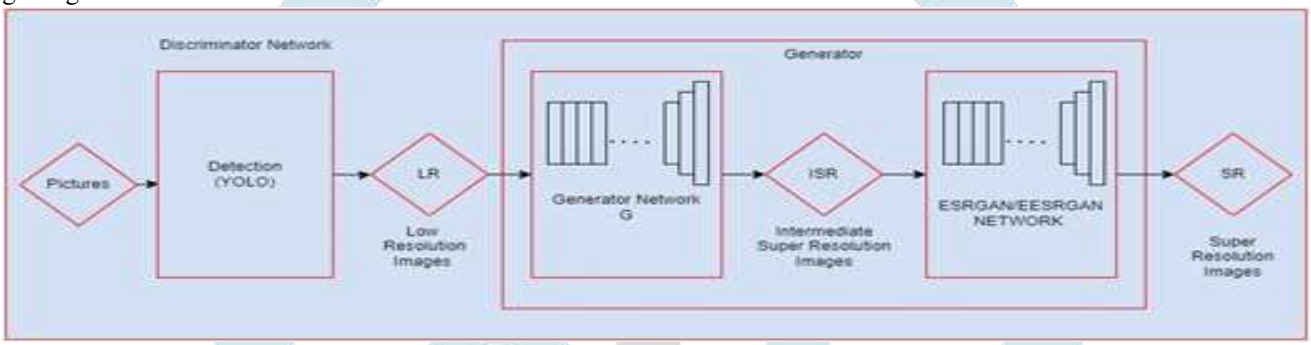
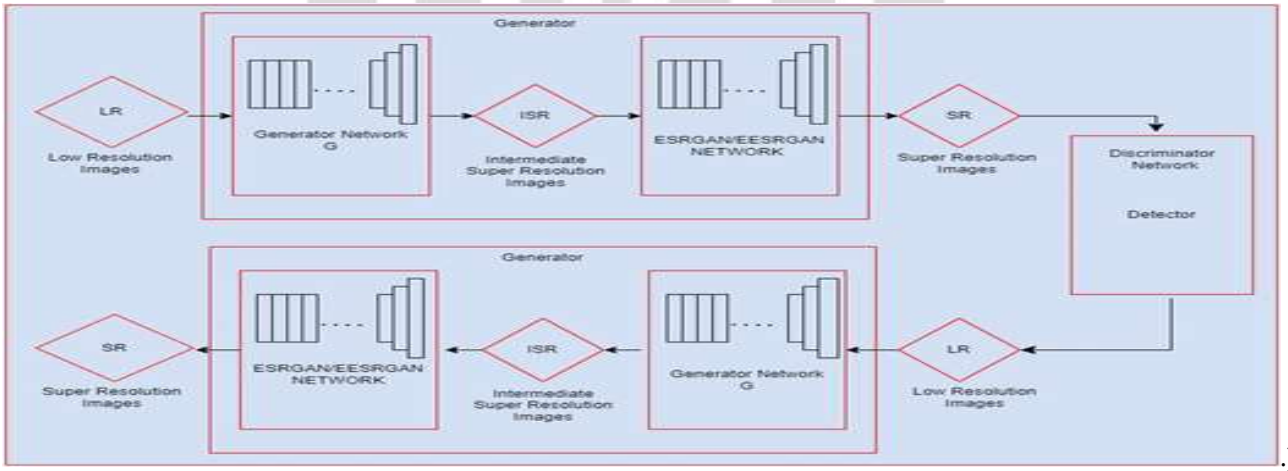


Fig. 5. network design as a whole includes a generator and a discriminator module.

In the second approach, low-resolution (LR) photos are processed via the use of a generator network to create superresolution images. Next, Intermediate Super Resolution Images ISR are produced, and additional augmentation is then carried out via the ESRGAN/EESRGAN network. The LR pictures are then again transformed into super resolution images using a Generator network, and a Discriminator Network is used for detecting purposes. The ESRGAN/EESRGAN network is then used to create and refine the ISR pictures in order to attain better resolution. .Fig.2 show network's architecture:



.)

Fig. 6. network design as a whole includes a generator and a discriminator module.

Fig. 8. LR images to SR images with Object detection using COCO dataset

network till convergence in different training sessions. The detector networks were then trained using the SR pictures.In order to determine the weights for from beginning to end training, we performed distinct exercises as a pre-training phase. Once the SR and identification of objects models have been learned simultaneously, the gradients from the object detector may then be transferred into the generator network.

We set the learning rate to 0.0001 and reduced it by half every 50,000 iterations throughout the training procedure. Five was chosen as the batch size. We used the Adam optimizer with b1 = 0.9 and b2 = 0.999 to improve the design. We adjusted the weights across the board till convergence. We utilized five RRDB blocks for the EEN network and 23 RRDB blocks for the generator G.We used the PyTorch framework to develop our architecture, and two NVIDIA Titan X GPUs were used for training and testing.

A. Datasets

1) COWC Datasets: The satellite images available in the COWC (Cars Overhead with Context) dataset [31] have a ground-level resolution of 15 cm per pixel. The dataset comprises images from six different areas, namely Toronto, Selwyn, Potsdam, Vaihingen, Columbus, and Utah, covering regions in Germany, the United States, and New Zealand. Our analysis primarily focused on datasets from Toronto and Potsdam.

For our dataset selection, we included a total of 32,716 images in which 12,716 distinct automobile instances. We exclusively used RGB photos for training and testing. To ensure standardized input, we utilized 256-by-256 pixel tiles, ensuring that each tile contained at least one automobile. Cars typically had lengths ranging from 24 to 48 pixels and widths between 10 to 20 pixels. Consequently, the automobile area in the tiles ranged from 240 to 960 pixels, which was relatively small compared to other prominent objects in satellite imagery.

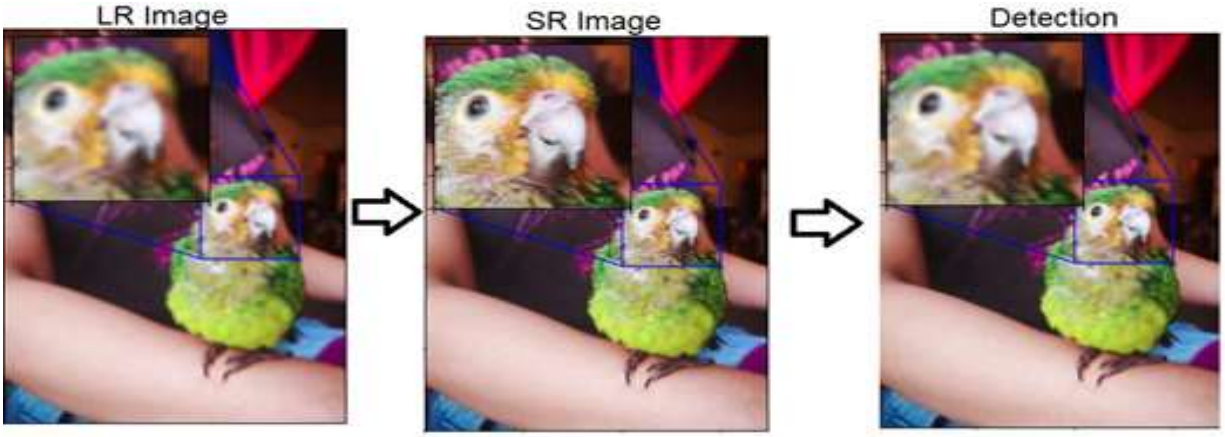
To generate low-resolution (LR) images, we employed bicubic downsampling with a downscale factor of 4 using the COWC dataset. This resulted in LR images with a resolution of 64 by 64 pixels. Each image tile was linked to a text database containing the boundary line dimensions for each car. Other object types were excluded from our experiments, focusing solely on the automobile class. Figure 6 displays examples taken from the COWC dataset.

A total of 3,340 tiles were used for training and testing. Following an 80%/20% split, the dataset was divided into a training set and a testing set. Furthermore, we partitioned the training set data into a training subset and a validation subset, with an 80%/20% split, respectively. To enhance the training process, we augmented the dataset by randomly rotating and flipping the images by 90 degrees.

Fig. 7. The COWC (Cars Overhead with Context) collection consists of pairs of low- and high-resolution photos, designated as (a, b), as well as images for object detection that show bounding boxes around cars, marked as (c).

2) COCO Datasets: COCO Train and COCO Val are the two primary picture datasets that make up the COCO dataset. While the COCO Train dataset comprises over 330k labelled pictures, the COCO Val dataset only has about 50,000. These photos include a broad variety of settings, things, and settings. Each picture in the COCO dataset has been carefully tagged with factual information. These annotations consist of bounding boxes that completely encompass the image's important objects. The collection furthermore offers pixel-level segmentation masks that precisely delineate the borders of items in the photos. 80 distinct object categories, including typical items like people, vehicles, pets, chairs, and more, are included in the COCO dataset. These groups reflect a wide range of items that are often seen in regular situations. Instance Segmentation: The COCO dataset offers instance-level segmentation masks in addition to bounding box annotations. These masks give each pixel in an item a distinct identifier, making it possible to analyze and comprehend object forms and limits in more depth. Caption Annotations: A portion of the photos in the COCO dataset also include caption annotations in addition to object annotations. These captions provide academics the opportunity to investigate picture captioning and natural language processing problems by providing descriptive textual information about the image content.

3) CIFAR-10 Datasets: The 60,000 RGB pictures in the CIFAR-10 benchmark [30] each have a size of 32 pixels by 32 pixels. There are ten categories in which these pictures are divided, including vehicles, birds, and airplanes. There are 6,000 photos total in each category, with 55,000 used for testing and 50,000 for training.We utilized the typical experimental conditions from [29] to enable a fair comparison with the work that is provided there, which focuses on lowresolution (LR) picture categorization. In these conditions, the first downsampling of the high-resolution (HR) photos by a factor of s = 0.25 produced images with a resolution of 8x8 pixels. The LR versions of the photos were then created by upscaling the downsampled images using nearest-neighbor (NN) interpolation to get them back to their original size of 32x32 pixels.



B. Evaluation Metrics for Detection

We received the results of our detection in the form of bounding boxes with corresponding class labels. To evaluate the accuracy of our conclusions, We devised measures including intersection over union (IoU), accuracy, and recall as well as the average precision (AP) metric for assessment.

In our analysis, we designated the collection of items that were successfully recognized as well as the assortment of items and true positives (TP) that were mistakenly detected as false positives (FP). The ratio of true positives to the total of true positives and false positives is used to determine precision. Further, we labeled the group of items that the detector missed as false negatives (FN). The proportion of identified objects (TP) to all items in the dataset is then used to determine recall.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (20)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (21)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (22)$$

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Precision + Recall

In this experiment, Our EESRGAN network and detectors underwent extensive training. The detectors and the discriminator DRa worked together to serve as a discriminator for the overall design. The SR network received the loss from the detectors and used it to improve the LR pictures. In the training phase, LR-HR picture pairings were utilized to train the EESRGAN component, and the detector was then trained using the resulting SR images. Only LR photos were fed into the network during testing. Our architecture performed object detection after creating an SR picture from the LR input.

The table ”Separate Training with Super-Resolution” displays the detection results obtained using various train/test pairs in conjunction with various super-resolution (SR) methodologies and detectors. It is clear that our innovative EESRGAN design produced the best results, coming close to detection rates reported when using just high-resolution (HR) pictures. Surprisingly, EESRGAN showed good results even when used directly on low-resolution (LR) images without access to HR data. Additionally, when employing alternative SR techniques for LR picture preprocessing, such as EESRGAN and ESRGAN, we saw considerable gains in average precision (AP).

Additionally, we included detectors in our comparison of our findings with other topologies, namely ESRGAN [21] and EESRGAN [22]. The comparative findings, as shown in the table, unmistakably show that our strategy surpasses the competition and yields better results.

TABLE I

TRAINED SR NETWORK IS USED FOR DETECTION ON SUPER-RESOLUTION (SR) PICTURES. BOTH LOW- AND HIGH-RESOLUTION (SR AND HR) PICTURES ARE UTILIZED TO TRAIN THE DETECTORS.

Model	Training Image Resolution-Test Image Resolution	COWC datasets (Test Result)	COCO Datasets (Test Result)	CIFAR-10 Datasets (Test Result)
ESRGAN +SSD	SR-SR	86.5%	82.2%	80.5%
	HR-SR	83.4%	79.4%	79.6%
ESRGAN +FRCNN	SR-SR	83.2%	81.2%	81.4%
	HR-SR	92.7%	80.3%	80%
ESRGAN +Yolo-4	SR-SR	96.2%	92.8%	93.2%
	HR-SR	95.4%	91.8%	91.9%

EESRGAN +SSD	SR-SR	87.1%	82.2%	81.9%
	HR-SR	84.3%	80.1%	80.3%
EESRGAN +FRCNN	SR-SR	94.3%	83.4%	82.3%
	HR-SR	93.9%	81.6%	81.7%
EESRGAN +Yolo-4	SR-SR	97.4%	94.8%	93.6%
	HR-SR	95.7%	93.8%	92.8%

Our approach is especially helpful for difficult applications, such unmanned aerial vehicle (UAV) and satellite remote sensing, since it can handle region-of-interests (RoIs) produced by any tiny object detector.

It is vital to investigate different datasets and investigate methods for producing more realistic low-resolution (LR) pictures. In conclusion, our method integrates many techniques to provide a better answer to the problem of finding tiny objects in LR images.

Future advances might make use of two different strategies.In the first method, an image is captured for detec.

Model	Training Image Resolution-Test Image Resolutio	COWC Dataset (Test Results)	COCO Dataset (Test Results)	CIFAR-10 Dataset (Test Results)
ESRGAN + SSD	SR-SR	89.5%	82.1%	80.9%
ESRGAN + FRCNN	SR-SR	94.6%	84.4%	82.2%
ESRGAN + Yolo-4	SR-SR	96.6%	95.7%	95.3%
EESRGAN + SSD	SR-SR	91.4%	83.7	81.6
EESRGAN + FRCNN	SR-SR	95.9%	85.6%	83.4%
EESRGAN + Yolo-4	SR-SR	97.8%	96.3%	96.5%

CONCLUSIONS

Our approach is especially helpful for difficult applications, such unmanned aerial vehicle (UAV) and satellite remote sensing, since it can handle region-of-interests (RoIs) produced by any tiny object detector.

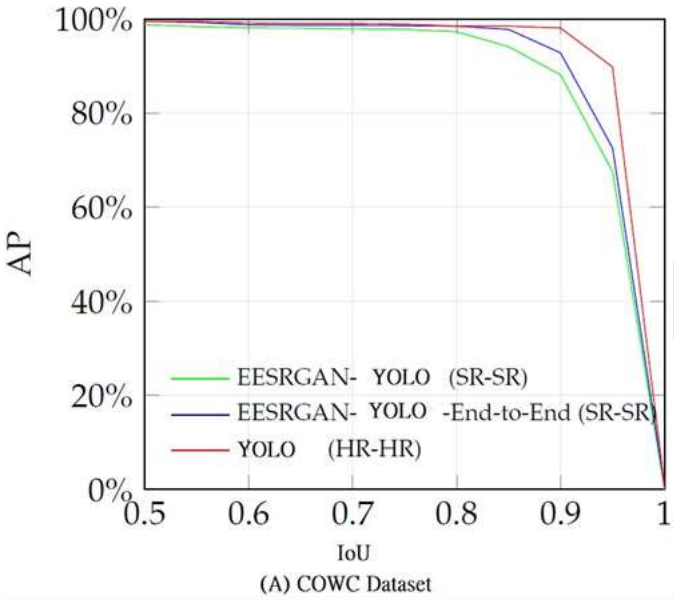
It is vital to investigate different datasets and investigate methods for producing more realistic low-resolution (LR) pictures. In conclusion, our method integrates many techniques to provide a better answer to the problem of finding tiny objects in LR images.

Future advances might make use of two different strategies.In the first method, an image is captured for detec-

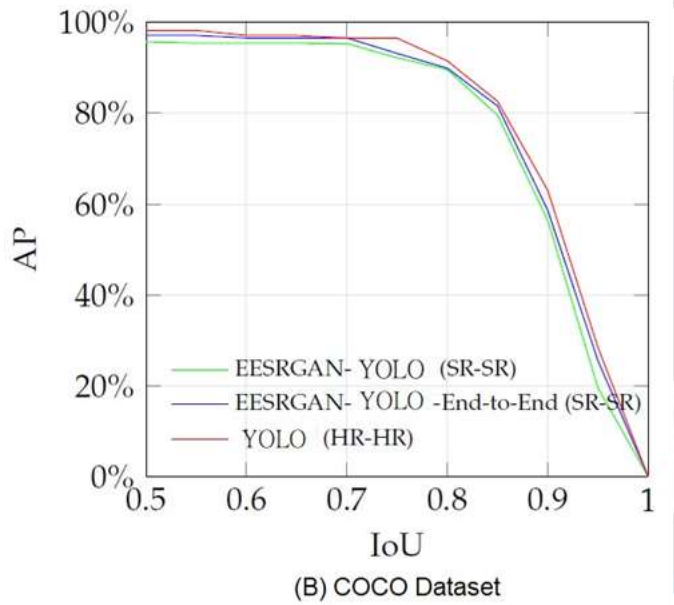
This study presents a complete architecture that uses lowresolution (LR) pictures as intake and produces object identification findings as outputs to allow the detection of things in satellite photography. The super-resolution (SR) network and the detector network are the two fundamental parts of the design. We have experimented with several pairings of superresolution (SR) algorithms and object detection strategies in order to assess the performance of the design. Using three separate datasets, we examined the average accuracy (AP) values for item identification. Through this investigation, we were able to evaluate how well the design worked with various SR algorithm and detection combinations. The results of the experiment show that, especially for the identification of tiny objects in satellite images, the recommended SR network paired with Yolo-4 produces the best results.

Convolutional neural network performance(CNN)-based approaches in applications employing low-resolution (LR) pictures has been shown to be negatively impacted by a problem with inconsistency between low-frequency (LF) and highfrequency (HF) components. We offer the GAN as a solution to this issue. GAN concurrently restores the lost data in both the LF and HF components

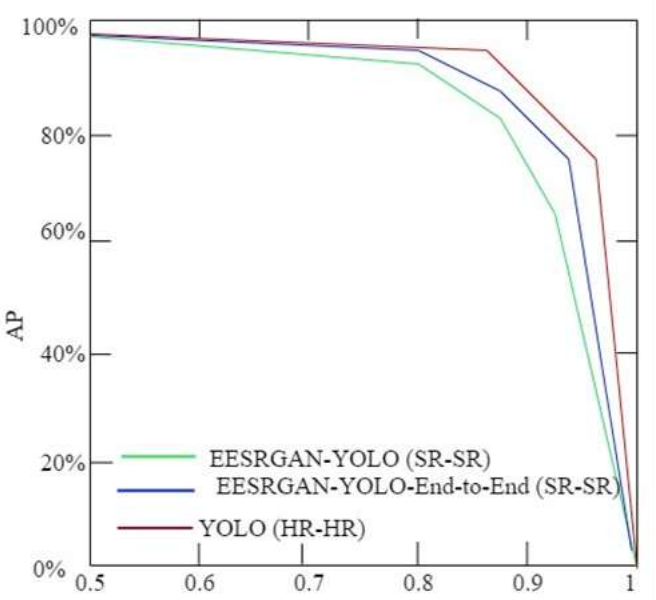
to provide enhanced picture



(A) COWC Dataset



(B) COCO Dataset



(C) CIFAR-10 Dataset

Fig. 10. The average precision-intersection over union (AP-IoU) datasets are shown by the curves in Figures (A), (B), and (C). These findings demonstrate the performance of our suggested technique on super-resolution (SR) pictures with and without end-to-end training, as well as the performance of solo Yolo-4 on high-resolution (HR) photos.

REFERENCE

[1] In 2020, TY. Li, X. Pei, Q. Huang, L. Jiao, R. Shang, and N. Marturi published ”Anchor-Free Single Stage Detector in Remote Sensing Images Based on Multiscale Dense Path Aggregation Feature Pyramid Network” in IEEE Access, number 8, pages 63121-63133. Access to the article is available at doi: 10.1109/ACCESS.2020.2984310. [CrossRef]

[2] Soft focal loss: Evaluating sample quality for dense object detection, Wang, Zhenyuan, et al. 271-280 in Neurocomputing 480 (2022).

[3] ”An End-to-End Adversarial Hashing Method for Unsupervised Multispectral Remote Sensing Image Retrieval,” 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, pp. 1536-1540, doi: 10.1109/ICIP40778.2020.9190949.

[4] Ewa Wilk, Magorzata Krowczyska, and Edwin Raczko. ”Recognition´ of asbestos roofs using high-resolution aerial photography and convolutional neural networks. evaluating many possibilities. 2022: 109092; Building and Environment 217.

[5] Jingwen Zuo et al., ”Research on image super-resolution algorithm based on mixed deep convolutional networks.” 95 (2021): 107422 in Computers and Electrical Engineering.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] ”Residual dense network for image restoration.” Zhang, Yulun et al. 2020: 2480–2495, IEEE Transactions on Pattern Analysis and Machine Intelligence, 43.7.

[8] Residual Dense Network for Image Super-Resolution. Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; Fu, Y. In the conference proceedings from Salt

Lake City, Utah, 18–23 June 2018, the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. [CrossRef]

[9] Yanghao Li et al., ”Mvitv2: Improved Multiscale Vision Transformers for Classification and Detection.” IEEE/CVF Conference on Computer Vision and Pattern Recognition Proceedings 2022.

[10] Pei-Wen Jian, Wei-Yen, and Hsu. For single picture super-resolution, detail-enhanced wavelet residual network. 2022: 1–13 IEEE Transactions on Instrumentation and Measurement.

[11] Song, Jie, et al. ”ESRGAN-DP: Enhanced super-resolution generative adversarial network with adaptive dual perceptual loss.” Heliyon 9.4 (2023).

[12] Liu, Zhiwei, et al. ”Stratified attention dense network for image superresolution.” Signal, Image and Video Processing (2022): 1-8..

[13] Meta-SR: A magnification-arbitrary network for super-resolution, X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan, and J. Sun, Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 1575–1584.

[14] Bosquet, Brais, et al. ”A full data augmentation pipeline for small object detection based on generative adversarial networks.” Pattern Recognition 133 (2023): 108998.

[15] Soufi, Omar, and Fatima Zahra Belouadha. ”Enhancing Accessibility to High-Resolution Satellite Imagery: A Novel Deep Learning-Based Super-Resolution Approach.” Journal of Environmental Treatment Techniques 11.2 (2023): 44-49.

[16] Ren, Kan, et al. ”Infrared small target detection via region super resolution generative adversarial network.” Applied Intelligence 52.10 (2022): 11725-11737.

[17] Wu, Yirui, et al. ”Edge computing driven low-light image dynamic enhancement for object detection.” IEEE Transactions on Network Science and Engineering (2022).

[18] KumarSingh, Nikhil, Nilay Laddha, and Joseph James. ”An Enhanced Image Colorization using Modified Generative Adversarial Networks with Pix2Pix Method.” 2023 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI). IEEE, 2023.

[19] Pham, Tuan D. ”Kriging-Weighted Laplacian Kernels for Grayscale Image Sharpening.” Ieee Access 10 (2022): 57094-57106. [CrossRef]

[20] ”Landsat8” Accessed on February11, 2020, at https://www.usgs.gov/land-resources/nli/landsat/landsat-8.

[21] Sentinel-2. Accessed on February11, 2020, from http://www.esa.int/Applications/Observing the Earth/Copernicus/Sentinel2.

[22] Shuhua Xu et al., ”A Positive-Unlabeled Generative Adversarial Network for Super-Resolution Image Reconstruction Using a Charbonnier Loss.” Signal 39.3 treatment (2022).

[23] Energy regulator for Alberta. (Accessed on February 5, 2020) AER is accessible online at https://www.aer.ca.

[24] Ding, Wei, Li Zhang, and Guangliang Yang. ”Building detection algorithm in multi-scale remote sensing images based on attention mechanism.” Evolutionary Intelligence (2023): 1-12. [CrossRef]

[25] Zhu, Zhengwei, et al. ”IRE: Improved Image Super-Resolution Based On Real-ESRGAN.” IEEE Access (2023).; [CrossRef]

[26] Jiang, J.; Jiang, K.; Wang, Z.; Yi, P.; Wang, G.; Lu, T. Edge-Enhanced GAN for Superresolution in Remote Sensing Images. 57, 5799–5812, IEEE Trans. Geosci. Remote Sens., 2019. [CrossRef]

[27] Shuhua Xu et al., ”A Positive-Unlabeled Generative Adversarial Network for Super-Resolution Image Reconstruction Using a Charbonnier Loss.” Signal 39.3 treatment (2022).

[28] Simonyan, Karen, and Andrew Zisserman. ”Very deep convolutional networks for large-scale image recognition [e-resource].” access mode: https://arxiv. org/pdf/1409.1556.–date of receipt 17 (2022).

[29] Bai, Dongxu, et al. ”Improved single shot multibox detector target detection method based on deep feature fusion.” Concurrency and Computation: Practice and Experience 34.4 (2022): e6614. [CrossRef]

[30] Ewa Wilk, Magorzata Krowczyska, and Edwin Raczko. ”Recognition´ of asbestos roofs using high-resolution aerial photography and convolutional neural networks. evaluating many possibilities. 2022: 109092; Building and Environment 217. [CrossRef]

[31] Singh, Gulbir, and Kuldeep Kumar Yogi. ”Performance evaluation of plant leaf disease detection using deep learning models.” Archives of Phytopathology and Plant Protection 56.3 (2023): 209-233.

[32] Jung, Hyun-Ki, and Gi-Sang Choi. ”Improved yolov3: Efficient object detection using drone images under various conditions.” Applied Sciences 12.14 (2022): 7255.

[33] Junos, M., Khairuddin, Anis, Thannirmalai, Subbiah ,Dahari, Mahidzal. (2021). Automatic detection of oil palm fruits from UAV images using an improved YOLO model. The Visual Computer. 38. 1-15.

10.1007/s00371-021-02116-3.

[34] Phutke, Shruti S., and Subrahmanyam Murala. ”Pseudo decoder guided light-weight architecture for image inpainting.” IEEE Transactions on Image Processing 31 (2022): 6577-6590.