



# Improving Tiny Object Detection in Aerial Images with YOLOv5

Ahmed Abdul-Hussain Sharba<sup>1\*</sup> , Hussein Kanaan<sup>2</sup> 

<sup>1</sup>Computer Engineering Department, College of Engineering, Mustansiriyah University, Baghdad, Iraq

<sup>2</sup>Faculty of Science, Lebanese University, Beirut, Lebanon

\*Email: [ahmedsharba@uomustansiriyah.edu.iq](mailto:ahmedsharba@uomustansiriyah.edu.iq)

## Article Info

**Received** 13/05/2024  
**Revised** 23/10/2024  
**Accepted** 01/11/2024

## Abstract

Object detection is a major area of computer vision work, particularly for aerial surveillance and traffic control applications, where detecting vehicles from aerial images is essential. However, such images often lack semantic detail and struggle to identify small, densely packed objects accurately. This paper proposes improvements to the You Only Look Once version 5 (YOLOv5) model to enhance small object detection. Key modifications include adding a new prediction head with a 160×160 feature map, replacing the Sigmoid Linear Unit (SiLU) activation function with the Exponential Linear Unit (ELU), and swapping the Spatial Pyramid Pooling – Fast (SPPF) module with the Spatial Pyramid Pooling (SPP) module. The enhanced model was tested on two datasets: Dataset for Object Detection in Aerial Images (DOTA) v1.5 and CarJet, which focused on vehicle and plane detection. Results showed a 7.1% increase in mean Average Precision (mAP) on the DOTA dataset and a 2.3% improvement on the CarJet dataset, measured with an Intersection over Union (IoU) threshold of 0.5. These architectural changes to YOLOv5 notably improve small object detection accuracy, offering valuable potential for aerial surveillance and traffic control tasks.

**Keywords:** Aerial Images; Computer Vision; Object Detection; You Only Look Once

## 1. Introduction

Object detection in aerial images is still being worked on; it has many uses, such as large-scale monitoring, intelligent transportation, and location-based services [1]. Recent work on the object detection problem has made much progress, but it is still challenging to solve when things in aerial images are tiny, like vehicles that are less than 8 pixels across [2]. Pixels in digital photographs encode color and intensity. Small object's limited pixel representation makes key aspects challenging to extract, whereas more oversized items provide a wealth of detail. Also, distinguishing small objects from background noise is complex in complicated settings [3]. The model must have high spatial resolution to locate and classify these objects among similar-sized parts and textures. To identify and analyze objects in aerial photographs, professionals in artificial intelligence utilize convolutional neural networks (CNNs) based on deep learning [4].

Object detection algorithms can be categorized into region-based and single-shot methods. Examples of region-based approaches consist of Mask Region-Based Convolutional Neural Networks (Mask R-CNN) [5], Faster Region-Based Convolutional Neural Networks (Faster R-CNN [6], and Fast Region-Based Convolutional Neural Networks (Fast R-CNN

[7]. Fast R-CNN proposes regions for training using an exhaustive selective search, then extracts features from these regions to classify data. Faster R-CNN, on the other hand, uses a Region Proposal Network (RPN) within a convolutional network to remove the need for intensive selective searching. An additional branch of pixel-to-pixel segmentation mask prediction is introduced by Mask R-CNN, which further expands Faster R-CNN. Single-shot techniques such as YOLO [8], Single Shot MultiBox Detector (SSD) [9], and various other ways have been introduced recently. These approaches see object detection as a regression task. YOLO processes the whole image in one step, predicting both bounding box and class probabilities concurrently. It utilizes objectness scores for making predictions. It is not advisable for small objects that make up a significant portion of the aerial information [10]. Compared to two-stage detectors, YOLO, which includes YOLOv5s, provides faster inferences and improves the detection accuracy for aerial images [11]. The YOLOv7, introduced in 2022, outperforms previous object identification models and earlier YOLO versions in terms of speed and accuracy. However, it still struggles with detecting small objects with a limited number of pixels due to the image resolution being reduced to 80×80 for such objects. This technology has various applications, such as monitoring

deforestation [12], tracking wildlife [13], disaster management [14], urban planning [15], precision agriculture [16], and maritime surveillance [17], demonstrating its significant impact across different fields.

The YOLOv5 algorithm consists of different versions. These are nano, small, medium, large, and X-large, based on the depth and width of the algorithm [18]. The YOLO algorithm is divided into three essential parts: the backbone, neck, and head [19]. The proposed algorithm EN-YOLO is based on YOLOv5s and provides a new prediction head of 160x160 resolution that significantly improves tiny object detection.

This paper aims to enhance object detection in aerial images, focusing on discovering tiny objects. It achieves its objectives by modifying the YOLOv5s architecture, including replacing the SPPF module with the SPP module, using the SiLU activation function instead of ELU, and adding a new prediction head.

## 2. Related Work

There is much work on the topic of object detection in aerial images. In 2017, Sommer et al. [20] introduced the detection of vehicles in aerial images using deep learning based on multi-category. The proposed approach is Faster R-CNN to detect objects based on multi-category. They implemented all experimental methods on DLR 3K Munich Vehicle Aerial Image, and the results are when method VGG16 was used, the result of AP for the cars and trucks was (58.8% and 30.6%) respectively. The mAP was (44.7%). The result of AP for the cars and trucks when adapted VGG16 was used is (65% and 27.5%), respectively, and the mAP was (46.2%). When the proposed Net method was used, the value of AP for cars and trucks was (91.6% and 25.3%) respectively, and the result of mAP was (58.8%).

In 2019, Yang et al. [21] introduced detecting clustered objects instead of individual objects. This approach addressed the challenges of tiny object pixels and objects' sparse and disorganized distribution. Using the ClusDet method with ResNet50, ResNet101, and ResNeXt101, the mAP results on the DOTA dataset were 32.0, 31.7, and 32.0, respectively.

In 2020, Su et al. [22] introduced a method for detecting multiscale key points in aerial images. Traditional keypoint-based detectors typically use a fixed-size feature map, which limits their ability to recognize objects of varying scales in aerial views. The authors proposed the multiscale keypoint detection network (MKD-Net) to address this issue. This novel network fuses multiscale layers to generate multiple feature maps for objects of different sizes. All feature maps can be used for corner prediction during the inference stage. The effectiveness of MKD-Net was evaluated using the mAP metric on two datasets: DOTA and PASCAL VOC. MKD-Net achieved 31.8 on the DOTA dataset and 44.8 on the PASCAL VOC dataset.

In 2021, Wang et al. [2] presented a method for detecting small objects in aerial images, those with a size of less than 8 pixels. The authors introduce a novel dataset called AI-TOD, which consists of 700,621 items categorized into eight distinct categories. The majority of the objects in this dataset have a size

of 12.8 pixels. The AI-TOD was constructed using publicly accessible extensive aerial image datasets, including DOTA-v1.5, xView, VisDrone2018-Det, Airbus Ship, and DIOR. Their methodology uses a Learning Network called M-CenterNet in combination with Multiple Centre Points. The M-CenterNet is an anchor-free keypoint detector that utilizes several points to precisely locate the center of an object, hence enhancing the accuracy of detecting small objects. The proposed method achieved accuracies 14.5, 40.7, 6.4, 6.1, 15, 19.4, and 20.4 in terms of AP, AP<sub>0.5</sub>, AP<sub>0.75</sub>, AP<sub>very tiny</sub>, AP<sub>tiny</sub>, AP<sub>small</sub>, and AP<sub>medium</sub>, respectively. Superior to Yolov3 and other methods.

In 2022, Pandey et al. [10] proposed a method to enhance object detection for drone or unmanned vehicle images. The challenges addressed in this work are the accuracy of small object detection, class imbalance issues, and boosting performance by contextual information. The method used in this work improved MCNN to obtain a density map and RetinaNet to detect small objects, and the results were evaluated on the VisDrone dataset. The results of the AP metric values were (29.6 and 29.9) respectively.

In 2022, Singh and Munjal [23] introduced an enhanced version of the YOLOv5l p6 model to improve the detection of small objects in aerial images. They used the prior version of YOLOv5 -p6 as the baseline model, which consists of four prediction heads by default, ranging from (80x80) for small objects to (10x10) for extra-large objects. The baseline model includes a focus module at the beginning of the backbone network and the Spatial Pyramid Pooling (SPP) at the end. This work aims to improve the YOLOv5l architecture for detecting small targets by modifying its structure. A new layer for feature fusion was incorporated into the feature pyramid module of YOLOv5l, enhancing its performance in detecting tiny objects. The algorithm's performance was evaluated using the DOTA and VisDrone datasets, benchmark datasets for aerial images. To increase accuracy in detecting tiny objects, input images were used at a resolution of 1024x1024 rather than the default 640x640 in YOLOv5. The enhanced YOLOv5l achieved a mean Average Precision (mAP) of 0.386 on the DOTA dataset, surpassing the 0.371 mAP achieved by YOLOv5x. On the VisDrone dataset, the enhanced YOLOv5l attained an mAP of 0.452, compared to 0.317 mAP in YOLOv5x, using confidence thresholds ranging from 0.5 to 0.95.

In 2023, Deng et al. [24] introduced a lightweight version of YOLOv5. This work proposed a new feature fusion method called the Deep Feature Map Cross Path Fusion Network (DFM-CPFN) to enhance the semantic information of deep features. Replacing the minimum detection head with the maximum detection head is recommended to improve performance. The second part of the article is devoted to constructing a new VoVNet-based module that enhances the backbone network's feature extraction capabilities. The study concluded by making the network more lightweight without compromising detection accuracy, using the concept of ShuffleNetV2. Compared to the original method, LAI-YOLOv5s achieves 40.4 on the mAP@0.5 index using the VisDrone2019 dataset.

In this paper, the contributions are:

- The Sigmoid Linear Unit (SiLU) activation function commonly utilized in the convolutional layers of YOLO has been substituted with the Exponential Linear Unit (ELU) activation function. This modification is accompanied by integrating an additional convolutional layer across all components of YOLO. This adjustment addresses issues related to the rate at which the loss function converges during the training phase.
- The deployment of Spatial Pyramidal Pooling-Fast (SPPF) at the terminal section of the YOLO backbone architecture has been replaced with Spatial Pyramidal Pooling (SPP). This decision stems from recognizing that while SPPF offers expedited processing, its suitability for accurately detecting small-scale objects is suboptimal. By reverting to SPP, the objective is to enhance the network's capability to handle small objects effectively.
- Another prediction head has been introduced, particularly for detecting tiny objects. The default YOLOv5 architecture includes three prediction heads that extricate features from neck feature maps of different sizes: (20x20) for large objects, (40x40) for medium objects, and (80x80) for small objects. This new version adds a (160x160) prediction head to detect tiny objects better. This enhancement is designed to improve the network's ability to detect objects of varying scales, thereby increasing its effectiveness in object detection tasks.

The EN-YOLO model used in this work enhances localization accuracy and achieves notable performance improvements on the DOTA and CarJet datasets when evaluated using the mAP metric. It overcomes default YOLOv5 and most other work. Choosing YOLOv5 instead of the latest versions of YOLO for the contribution is due to YOLOv5 offering specialized anchor box customization, stability, compatibility, performance adequacy, community support, and suitability for detecting very small objects in aerial images. In addition, the last versions need more considerable computation resources without a significant difference in accuracy.

### 3. Materials and Methods

#### 3.1. Dataset

Two types of datasets are used. The first one is the large-scale Google Earth images (DOTA), which contain 2806 images with 15 classes. The images include objects with a diverse range of sizes, orientations, and shapes, and each one is approximately 4000×4000 pixels in size[25]. Only photos of vehicles and plane objects were collected, about 1325 images.

The DOTA dataset contains about 52763 small objects (less than 3 pixels) of 145195 objects, as shown in Fig. 1. This number is problematic for authors studying object detection.



**Figure 1.** Sample of DOTA dataset images[25]

Another dataset, CarJet, was compiled from multiple sources, including the HRPlane, Semantic Drone, and Aerial Car datasets. It consists of approximately 2,043 images captured by drones and unmanned aerial vehicles (UAVs) from low altitudes. Consequently, the majority of objects in these images are medium to large, as depicted in Fig. 2.



**Figure 2.** Sample of CarJet dataset images

Acknowledging that using only two classes (vehicles and planes) reduces the data's diversity and may simplify the classification task is essential. However, the primary objective is to develop a model that can reliably distinguish between vehicles and planes, the most common objects of interest in target urban scenarios. Although these two classes may not frequently co-occur in cities, this approach aims to address scenarios where distinguishing between these broad categories is critical. This focus on vehicles and planes is based on their relevance to urban monitoring and surveillance applications. In this work, small and large vehicles have been combined into a single "vehicle" class. The primary reason for this decision was to simplify the classification problem and focus on the broader category of vehicles commonly found in urban environments. By doing so, the objective is to develop a generalized model capable of detecting vehicles without distinguishing between different subcategories.

### 3.2. Labeling

All objects in the images must be labeled manually before beginning the training in the YOLO algorithm, which assigns a class to each object called the ground truth [26].

All images in two datasets were labeled manually by an application called "lebelImg," as shown in Fig. 3. The "lebelImg" app provides advanced labeling tools, allowing precise labeling of various object shapes and sizes. To minimize labeling errors, it includes built-in quality control mechanisms, such as label validation and consistency checks. Additionally, a validation process was conducted in which multiple annotators manually reviewed a subset of the labeled data to ensure consistency and accuracy.



Figure 3. lebelImg application

### 3.3. Overview of YOLOv5

The YOLO algorithm idea is different from other systems because it predicts both surrounding boxes and classes at the same time. First, the picture sent is split into a ( $S \times S$ ) grid. Next,  $B$  bounding boxes are set up in each grid cell, and each one has a confidence score [27]. In this case, "confidence" means the chance that an object is inside each box, as shown in Equation (1):

$$C = P * IOU \quad (1)$$

IOU, which stands for intersection over union, is a fraction containing values between 0 and 1. Fig. 4 illustrates the concept of intersection, representing the area where the predicted bounding box overlaps with the ground truth. The union refers to the combined area covered by both the predicted and ground truth bounding boxes. Ideally, the Intersection over the Union (IOU) should be close to 1, indicating that the predicted bounding box closely matches the actual bounding box [28]. The ( $P$ ) is a prediction bounding box, and the ( $C$ ) is a confidence score, so if the ( $C$ ) is equal to or greater than the threshold of a specific class that is determined in training (default is 0.5), the object assigned to this class. Equation (2) describes the IOU.

$$IOU = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

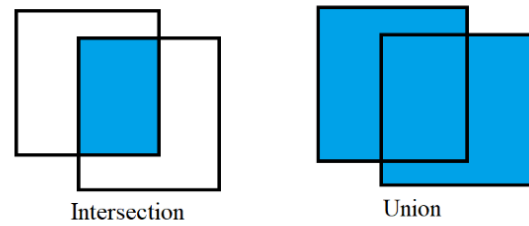
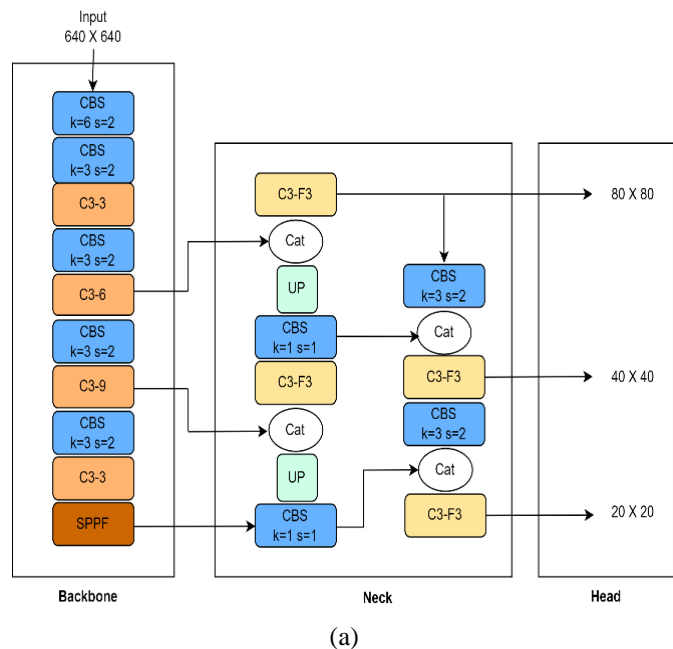


Figure 4. Concept of Intersection and Union

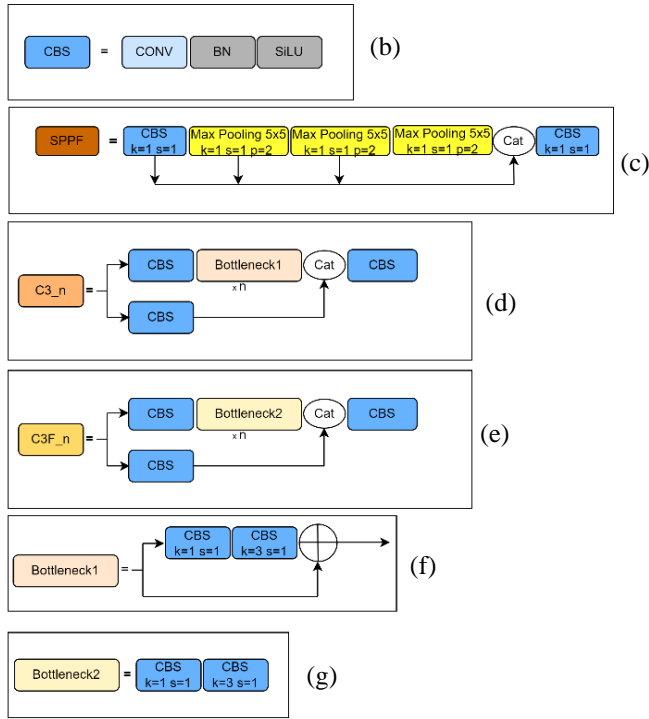
The YOLOv5 model is structured into three key components: the Backbone, Neck, and Head modules.

- The Backbone network is responsible for extracting feature information from input images.
- The Neck module integrates these features to generate three feature maps at different scales.
- The Head module then utilizes these feature maps to detect objects [29].

The backbone network of YOLOv5 is primarily constructed using the CSPDarkNet53 architecture, which incorporates Convolutional (Conv), C3, and SPPF layers. Convolution, batch normalization, and the SiLU function are all components that together make up the Conv layer. The C3 module can minimize the number of model parameters by utilizing residual connections, ultimately increasing the inference speed. The SPPF module consists of three max-pooling  $5 \times 5$  layers. YOLOv5 utilizes the Path Aggregation Network (PANet) in its Neck component, which enhances the Feature Pyramid Network (FPN) with bottom-up paths. Following top-down feature fusion in FPN, the bottom-up routes convey positional information from lower layers to deeper ones, significantly improving the localization capability across various scales [30]. The YOLOv5 architecture is shown in Fig. 5.







**Figure 5.** (a) The YOLOv5 Architecture (b) The CBC (c) The SPPF (d) The C3-True (e) The C3-False (f) The Bottleneck 1 (g) The Bottleneck 2

### 3.4. Evaluation Metrics

This work assesses the performance of the original YOLOv5 model and EN-YOLOv5s models' performance using several metrics, including Precision, Recall, F1-score, Average Precision (AP), and Mean Average Precision (mAP). Precision (P) is calculated as shown in Equation (3), Recall (R) is given by Equation (4), and the F-Score is defined in Equation (5). The Average Precision (AP) is computed as the integral of Precision over Recall, as shown in Equation (6), and the Mean Average Precision (mAP) is the mean of AP values across all classes, as expressed in Equation (7).

$$Precision(P) = \frac{Tp}{Tp+FP} \times 100 \quad (3)$$

$$Recall\ Rate\ (R) = \frac{Tp}{Tp+Fn} \times 100 \quad (4)$$

$$F - Score\ (\%) = 2 \times \frac{(P \times R)}{(P+R)} \times 100 \quad (5)$$

$$AP = \int_0^1 P(R) dR \quad (6)$$

$$AP = \frac{\sum_{i=1}^n AP(i)}{n} \quad (7)$$

True Positive (TP) represents the accurately detected knots, False Positive (FP) signifies the additional knots incorrectly identified on the timber surface (commission error), and False Negative (FN) indicates the missed knots during detection (omission error). Precision is the proportion of accurately identified knots among all the knots that were identified. The Recall Rate served as a measure of the detector's sensitivity. The F-Score offered a method to merge Precision and Recall

Rate into a unified score encompassing both aspects. A higher F-Score signifies a more precise model.

## 4. The Proposed Method (EN-YOLOv5s)

There are many challenges with identifying tiny objects; therefore, many authors are trying to figure out a solution. The proposed contributions are:

### 4.1. Activation Function

The Convolutional layer commonly uses the Rectified Linear Unit (ReLU) as its activation function due to its fast learning and simple implementation, owing to its low computational demands. However, a limitation of the ReLU activation function is that when it produces a value less than zero, the gradient remains at zero, which can cause the weight to stay zero throughout the training process. As a result, this can hinder effective learning. The ReLU function is defined in Equation (8), and its derivative is shown in Equation (9).

$$ReLU(x) = \max(0, x) \quad (8)$$

$$ReLU'(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

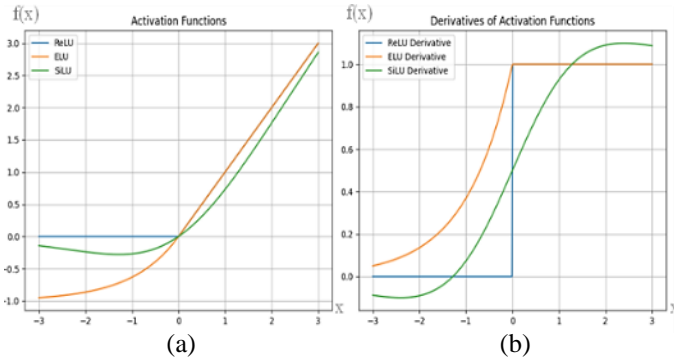
The ELU activation function is an altered version of the ReLU activation function. As a result, the training time is reduced, and neural networks' performance on the test set is improved. When  $x$  is less than zero, the exponential function links the differential function without breaking. A broken function, like the step function, might lead to local optima, as shown in Fig. 6 since the loss function can be constructed unevenly. Equations (10) and (11) describe the ELU activation function and its derivative, respectively.

$$ELU(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha e^x - 1 & \text{if } x \leq 0 \end{cases} \quad (10)$$

$$ELU'(x) = \begin{cases} 1 & \text{if } x > 0 \\ f(x) + \alpha & \text{if } x \leq 0 \end{cases} \quad (11)$$

$\alpha$  is typically defined as 1. (If  $\alpha$  does not equal 1, it is called SeLU.) However, the exclusive linear unit combines the benefits of ReLU and tackles the issue of Dying ReLU. The exponential function is computed differently from the normal ReLU, and the output value is almost perfectly centered around zero.

The comparison between the three activation functions is shown in Fig. 6.



**Figure 6. a. SiLU, ReLU, and ELU Activation Function  
b. SiLU, ReLU, and ELU Derivative Function**

These issues can be addressed by using the Sigmoid Linear Unit (SiLU) activation function. SiLU, as defined in Equation (12), becomes saturated with negative inputs, which can sometimes cause gradients to vanish and impede learning. Moreover, SiLU is generally limited to systems based on reinforcement learning and is typically only accessible in the hidden layers of deep neural networks. The derivative of the SiLU function is presented in Equation (13).

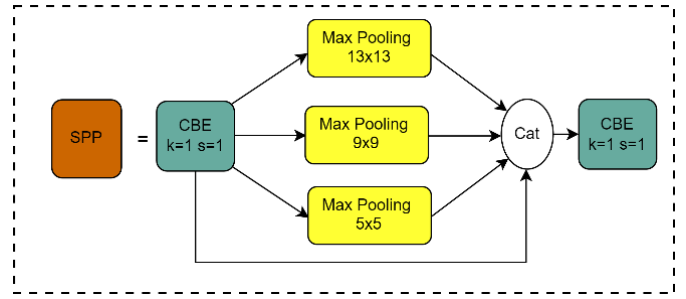
$$\text{SiLU}(x) = x * \frac{1}{1+e^{-x}} \quad (12)$$

$$\text{SiLU}'(x) = \frac{1}{1+e^{-x}} + \left\{ \left( x * \frac{1}{1+e^{-x}} \right) * \left( 1 - \frac{1}{1+e^{-x}} \right) \right\} \quad (13)$$

The ELU activation mechanism is employed to resolve this all-encompassing issue. The SiLU and ELU activation functions both address the issue of dying ReLU. However, the SiLU function has limited applicability, leading us to substitute it with the ELU function in all parts of YOLOv5s.

#### 4.2. Spatial Pyramid Pooling (SPP)

The most recent version of YOLOv5 uses SPPF, which alters the pooling core size to be the same for all of the cores and replaces the three parallel max pooling in SPP with serial pooling. By optimizing the process of pooling, it is possible to avoid duplicating SPP processes, which in turn helps to enhance the speed at which the network operates. Even while SPPF increases the network's detection rate, it is not optimal for detecting small dense objects (such as the objects in the DOTA dataset) due to its imperfect accuracy, so we suggest returning to SPP. In the YOLOv5 network, the Spatial Pyramid Pooling (SPP) structure is designed to generate a fixed-size feature vector from images of varying dimensions as output from the fully connected layer. To enhance the network's receptive field, improve the feature map's expressive capability, and leverage the maximum pooling operation for feature extraction, the SPP structure uses convolutional kernels of sizes 5, 9, and 13. It starts by performing  $(1 \times 1)$ ,  $(5 \times 5)$ ,  $(9 \times 9)$ , and  $(13 \times 13)$  maximum pooling operations in parallel on the data processed through the convolutional normalization activation function. These results are then concatenated and combined with the CBE structure[31]. The architecture of the SPP module is illustrated in Fig. 7.



**Figure 7. The Structure of The SPP Module**

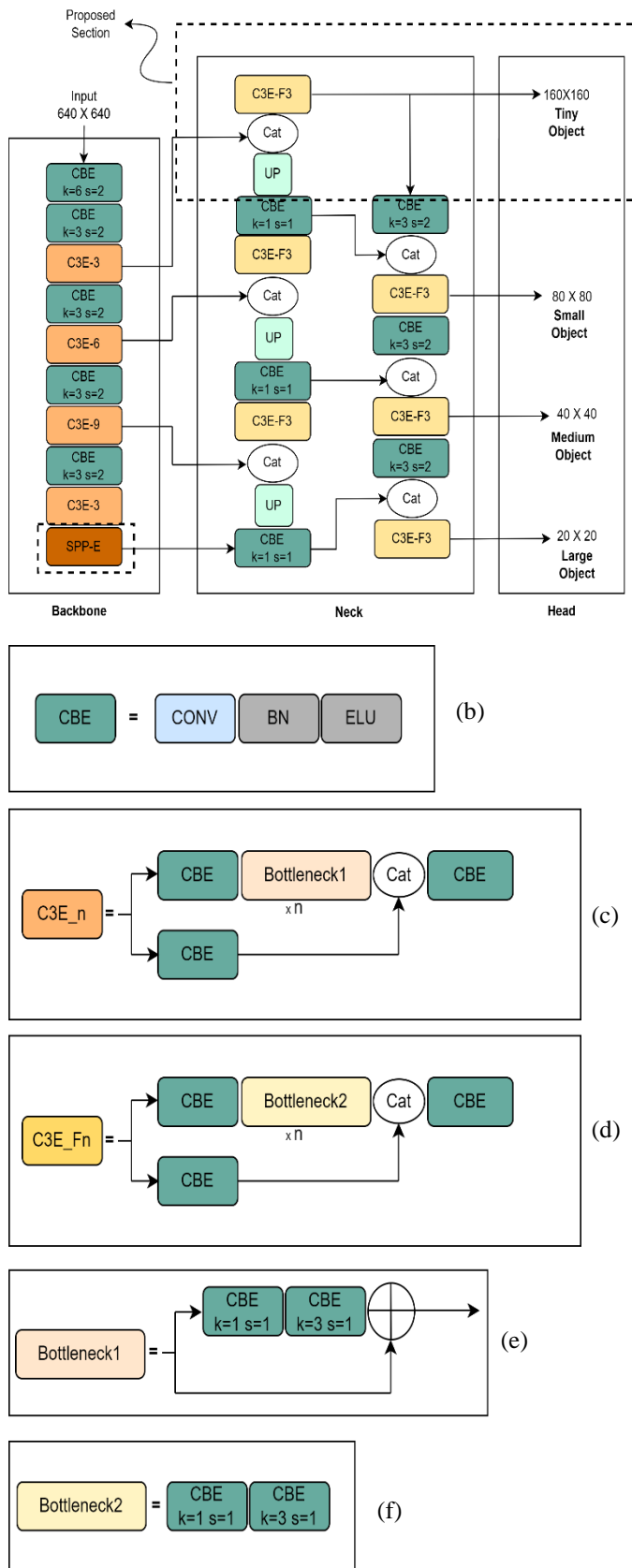
Spatial Pyramid Pooling (SPP) is beneficial in the sense that it is best for small object detection tasks in terms of capturing multi-scale features, adaptively adjusting the receptive field of the network, including contextual information, being invariant to scale, being tolerant of variation in size and aspect ratio, and rectifying the issue of information loss. For all these reasons, SPP is essential in any object detection network, especially one that deals with precise small object detection.

#### 4.3. Prediction Head

The YOLOv5s network can only handle down-sampling steps up to 32. So, a target is considered small if its resolution is less than  $32 \times 32$  pixels, large if its size is more than  $96 \times 96$  pixels, and medium if its resolution is in the middle. Two additional categories are created for targets whose resolution is less than  $32 \times 32$  pixels: tiny (resolution  $< 16 \times 16$  pixels) and small ( $16 \times 16$  pixels  $<$  resolution  $< 32 \times 32$  pixels), due to the high number of small-scale objects in UAV images.

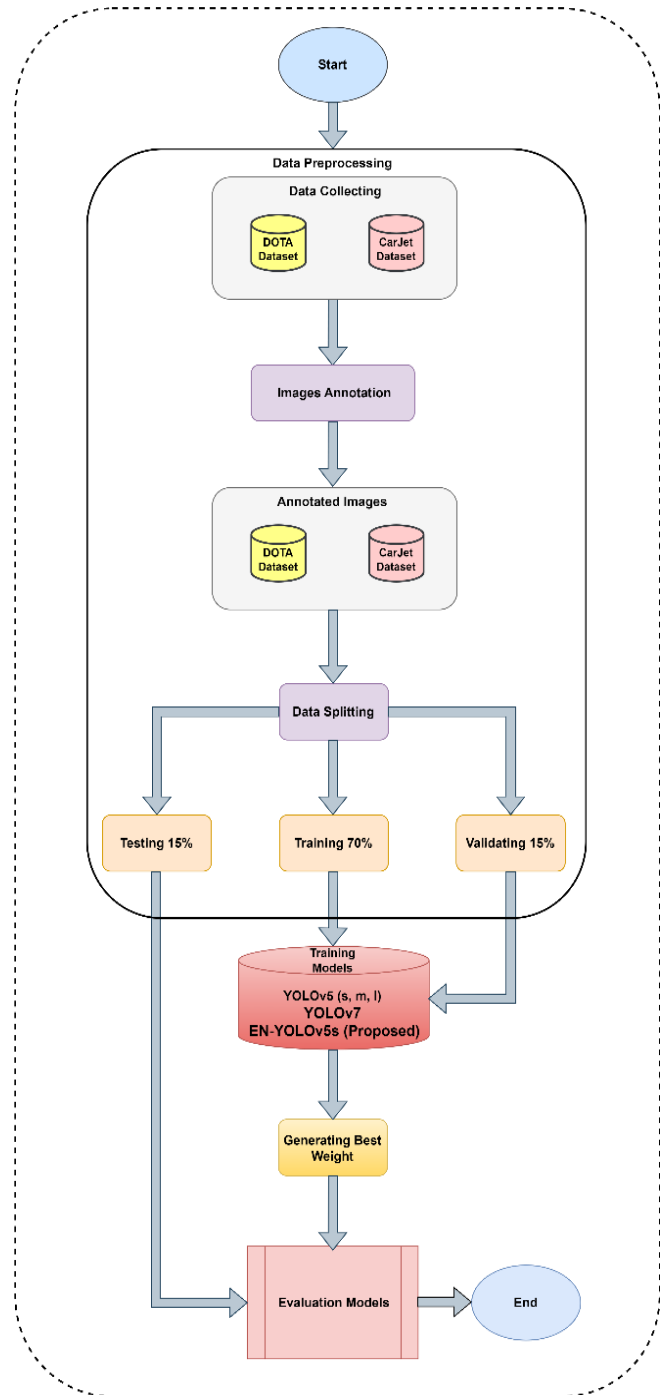
The YOLOv5s network features three detection layers: P3, P4, and P5. These layers handle feature maps with sizes of  $80 \times 80$ ,  $40 \times 40$ , and  $20 \times 20$ , respectively. Larger feature maps are better suited for detecting smaller objects. The  $80 \times 80$  feature map corresponds to an input size of  $640 \times 640$ ; each grid has an  $8 \times 8$  reception area. The network has trouble identifying a small object's features when its height or width in the image is less than 8 pixels. The new P2 detection branch addresses this by detecting objects as small as  $4 \times 4$  pixels, using smaller anchor boxes to reduce the number of missed detections for small vehicles significantly.

According to Fig. 8, the initial C3 module in the Backbone generates a feature map with dimensions of  $160 \times 160$  after undergoing two down-samplings. In contrast, the second C3 module produces a feature map with dimensions of  $80 \times 80$ . The P2 detection branch is created by combining the up-sampled output from the second C3 module with the  $160 \times 160$  feature map. The shallow convolutional layer, which contributes extra shape, location, and size information, is the main input source for P2. This enriched data helps the model better distinguish subtle features, thereby improving its ability to detect small targets more accurately. The overall EN-YOLOv5s are shown in Fig.8.



**Figure 8.** (a) EN-YOLOv5 Architecture (b) The CBE (c) The C3E-True (d) The C3E-False (e) The Bottleneck1 (f) The Bottleneck2

EN-YOLOv5s gains more accuracy than most other algorithms. However, it may still encounter limitations inherent to the YOLO framework, including potential challenges in handling complex scenarios with occlusions, variations in lighting conditions, and diverse object scales. Furthermore, EN-YOLO's performance could degrade when implemented on datasets with different characteristics from its development environment. Fig. 9 shows the proposed method's flow chart diagram.



**Figure 9.** The Flowchart of the proposed method

## 5. Experiment

Experiments on two aerial picture datasets, DOTA and CarJet, validated the proposed model's performance.

### 5.1. Experimental Environment

The suggested model was implemented using the official YOLOv5 GitHub repository. The model was modified and then trained using Python Jupyter notebooks on the Google Colab Pro platform.

### 5.2. Experimental Work

Both datasets were trained and validated for 100 epochs on the default YOLOv5s, YOLOv5m, YOLOv5l, YOLOv7, and the proposed model EN-YOLOv5s in several stages with a learning rate of 0.01 and optimizer SGD.

## 6. The Results

Three modifications are proposed to contribute to improving the detection of small objects. Firstly, a new head prediction for tiny object detection, and then the activation function is replaced with ELU. At last, replace the SPPF with the SPP module. The results are obtained at every step for the two datasets and compared with the default structure of YOLOv5 (S, M, and L), YOLOv7, and other state-of-the-art methods. The training, validation, and testing results illustrate the efficiency of EN-YOLOv5s compared with all versions of YOLOv5 and YOLOv7 in both datasets. Table 1 shows the AP and mAP for the DOTA dataset and Table 2 for the CarJet dataset.

**Table 1.** Comparison of mAP for different structures of YOLOv5 for the DOTA dataset

Methods	mAP	mAP (0.5)%
YOLOv5s	48.9	28.9
YOLOv5m	49.4	31.4
YOLOv5l	50.5	33.5
YOLOv7	51.3	32.6
YOLOv5s+P2+SiLU+SPPF	53.2	31.7
YOLOv5s+P2+ELU+SPPF	53.9	34.3
<b>YOLOv5s+P2+ELU+SPP (Proposed)</b>	<b>54.4</b>	<b>35.1</b>

The proposed model is better than the others for both metrics. The proposed model achieved progress of about (7.1) in terms of mAP (0.5-0.95) and (6.7) in terms of mAP (0.5) from YOLOv5s, which is the same version. Also, the proposed model overcomes the medium and large versions of YOLOv5, and the results showed that the ELU activation function and the SPP module are better for tiny and dense objects.

**Table 2.** Comparison of mAP for different structures of YOLOv5 for the CarJet dataset

Methods	mAP	mAP (0.5)%
YOLOv5s	95	70.2
YOLOv5m	96.2	71.2
YOLOv5l	96.7	72.3
YOLOv7	96.6	72.3
YOLOv5s+P2+SiLU+SPPF	95.3	70.3
<b>YOLOv5s+P2+ELU+SPPF (Proposed)</b>	<b>96.9</b>	<b>72.5</b>
YOLOv5s+P2+ELU+SPP	94.7	68.9

Table 2 shows that the SPPF module is better for medium and large objects than SPP.

Table 3 illustrates the DOTA dataset's precision and recall metric value and Table 4 for the CarJet Dataset.

**Table 3.** Comparison of Precision/Recall for different structures of YOLOv5 for the DOTA dataset

Methods	Precision	Recall
YOLOv5s	80	41.1
YOLOv5m	80.2	42.3
YOLOv5l	<b>84.9</b>	42.4
YOLOv7	84.5	<b>48.9</b>
YOLOv5s+P2+SiLU+SPPF	80.1	45
YOLOv5s+P2+ELU+SPPF	80.8	46.3
YOLOv5s+P2+ELU+SPP (Proposed)	82.8	48.8

Here, the YOLOv5l model achieves gain compared to another approach regarding the Precision metric, and the YOLOv7 model achieves gain compared to another approach regarding the Recall metric.

The result in Table 4 also shows that the proposed approach with the SPPF is better than the other in terms of the precision metric, and the YOLOv7 model achieves a gain compared to another approach regarding the Recall metric.

Table 5 compares the proposed EN-YOLOv5s model with state-of-the-art methods to demonstrate its effectiveness.



**Table 4.** Comparison of Precision/Recall of different structures of YOLOv5 for the CarJet dataset

Methods	Precision	Recall
YOLOv5s	95.4	94.2
YOLOv5m	96.5	94.3
YOLOv5l	96.9	94.7
YOLOv7	95.7	<b>96.1</b>
YOLOv5s+P2+SiLU+SPPF	95.6	94.4
<b>YOLOv5s+P2+ELU+SPPF (Proposed)</b>	<b>97.9</b>	95.1
YOLOv5s+P2+ELU+SPP	94	93.9

**Table 5.** Validation Comparison of The Proposed model (EN-YOLOv5s) with the five Recent Different Models for the DOTA dataset

Methods	mAP	mAP (0.5)%
ClusDet (ResNet50)[21]	47.6	32.0
ClusDet (ResNet101)[21]	47.8	31.7
ClusDet (ResNeXt101)[21]	47.1	32.0
MKD-NET[22]	53.8	31.8
YOLOv5imprv[23]	<b>60.3</b>	<b>38.6</b>
<b>EN-YOLOv5s (Proposed)</b>	54.4	35.1

Table 5 shows that the proposed model outperformed the first four models in terms of mAP metrics. However, the YOLOv5imprv model achieved higher accuracy than the proposed model. This is attributed to using YOLOv5l as a baseline model and an input image resolution of 1024x1024, which directly impacts accuracy.

Table 6 illustrates the parameters and Giga Floating Point Operations (GFLOPs) of EN-YOLOv5s. Where the parameters refer to the weights and biases in the neural network. In the context of YOLOv5, these parameters are learned during training and are crucial for the model's ability to make accurate predictions. The number of parameters in a model is often used to measure its complexity; more parameters typically allow the model to capture more intricate patterns in the data but also require more computational resources for training and inference. GLOPs represent the number of floating-point operations (FLOPs) required to make a prediction, measured in billions. This metric is a measure of the computational complexity of the model. A higher number of GLOPs indicates that the model requires more computational power to process an input. In the context of YOLOv5, GLOPs are used to assess the efficiency and speed of the model during inference.

**Table 6.** Comparison of Parameters/GFLOPs of different structures of YOLOv5

Methods	Parameters	GFLOPs
YOLOv5s	7025023	16
YOLOv5m	20875359	47.9
YOLOv5l	46143679	108.2
YOLOv7	37200095	105.4
YOLOv5s+P2+ELU+SPP (Proposed)	8396820	40

Table 6 shows that the parameters and GFLOPs of the proposed model are increased compared with the original YOLOv5s but still less than YOLOv5m, YOLOv5l, and YOLOv7 with better accuracy.

Although the accuracy increased, the model's speed decreased, so when a new predicted head with a resolution of 160x160 was added to the model, the parameters and GFLOPs increased, too.

Fig. 10 illustrates the curve of Precision/Recall of the proposed model for the DOTA dataset.

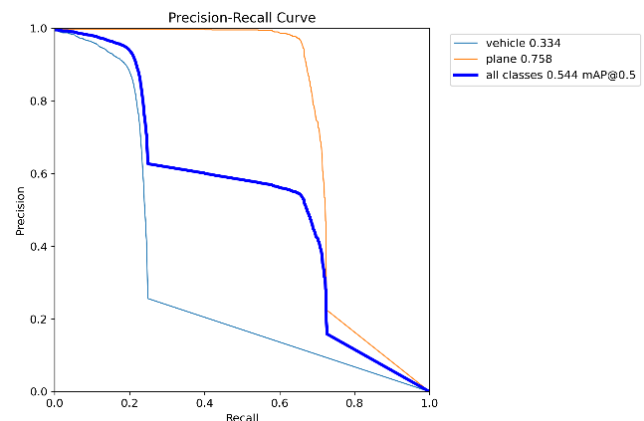
**Figure 10.** Precision/Recall curve for DOTA dataset

Fig. 11 illustrates the curve of Precision/Recall of the proposed model for the CarJet dataset.

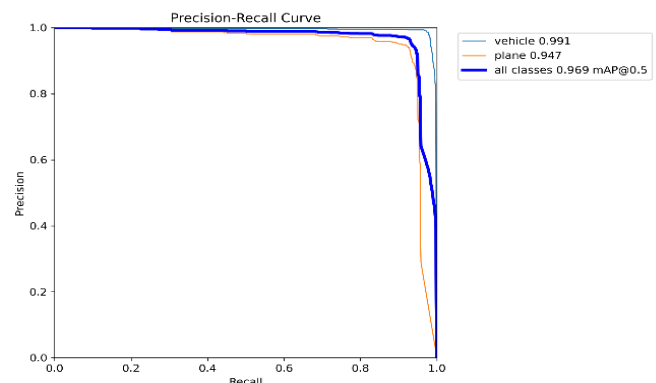
**Figure 11.** Precision/Recall curve for CarJet dataset

Fig. 12 shows a sample image for the test of EN-YOLOv5s.



**Figure 12.** Sample Image for The Test of EN-YOLOv5s

Fig. 12 shows a good detection of small objects using the proposed model.

The result illustrates that the proposed model is better than others for the DOTA dataset. Although the SPP module consumes more resources, it is better for tiny clustered objects than SPPF modules. This is in contrast with another dataset, which contains larger objects. The ELU activation function is better than others, and adding a prediction head helps improve the detection of tiny objects.

Fig. 13 shows a sample image for the failed detection of two planes on EN-YOLOv5s.



**Figure 13.** Sample Image for The Failed Detection on EN-YOLOv5s

## 7. Conclusions

This work is based on the modified YOLOv5s architecture, significantly improving object detection, especially for small and clustered objects. The results show substantial advancements in detecting challenging objects, with a remarkable increase in the overall accuracy of mAP scores for small object classes in the DOTA dataset. Additionally, detection performance improves when using the SPPF module on the CarJet dataset. This work illustrates the efficiency of the proposed model compared with YOLOv7 and most state-of-the-art algorithms.

The work findings strongly support the effectiveness of the proposed model improvements in enhancing object detection accuracy. Although these modifications led to some unexpected slowdowns in model speed, the trade-offs are valid as they significantly advance detection performance.

The broader implications of this work include progress in aerial image analysis for surveillance and traffic management applications, addressing practical problems by improving object detection accuracy.

However, the work has limitations, such as potential sample size and methodology challenges. Additional studies with various datasets and scenarios are necessary to validate our findings further. Future work should aim to optimize model architectures for a better balance between accuracy and efficiency and explore different methods of object detection in aerial images.

## 8. Future Work

In future work, the focus will include computing the capacity of parking areas and airports to build an effective IoT light model. By integrating the enhanced object detection algorithms into IoT frameworks, we can develop systems for real-time monitoring and management of parking spaces, airport logistics, traffic control, and surveillance. This will involve:

- **Parking Capacity Computation:** Developing algorithms to accurately detect and count vehicles in parking lots,
- enabling real-time updates on available parking spaces and efficient parking management.
- **Airport Capacity Management:** Applying object detection techniques to monitor airplane positions and movements within airport premises, optimizing ground operations, and enhancing safety protocols.
- **IoT Light Model Integration:** Designing lightweight IoT systems that leverage the improved object detection capabilities to provide real-time data and analytics for smart city applications, including traffic management and surveillance.
- **Traffic Control and Surveillance:** Enhancing traffic control systems by providing accurate real-time data on vehicle movement, congestion patterns, and potential incidents, thereby improving response times and traffic flow management.

These advancements aim to contribute to the development of smart infrastructure and enhance the efficiency and effectiveness of urban management systems. Improving data collection methodologies and analytical techniques will ensure robust and reliable object detection algorithms. This work contributes valuable insights into aerial surveillance and traffic control through advancements in object detection algorithms. Its objective is to guide future work and applications in aerial image analysis.

## Acknowledgments

The writers are grateful to Mustansiriyah University and Art, Science, and Technology University of Lebanon and (<https://www.uomustansiriyah.edu.iq/>) and (<https://www.aul.edu.lb/>)

## Conflict of interest

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

## Author Contribution Statement

Ahmed Abdul-Hussain Sharba: problem statement, proposed the model, and generated the results.

Hussein Kanaan developed the model and supervised the findings of this work.

Both authors contributed to the analysis of the results.

## References

- [1] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, "Object detection in optical remote sensing images: A survey and a new benchmark," *ISPRS journal of photogrammetry and remote sensing*, vol. 159, no. 1, pp. 296–307, 2020, <https://doi.org/10.1016/j.isprsjprs.2019.11.023>.
- [2] J. Wang, W. Yang, H. Guo, R. Zhang, and G. S. Xia, "Tiny object detection in aerial images," in *Proceedings - International Conference on Pattern Recognition*, Institute of Electrical and Electronics Engineers Inc., 2020, pp. 3791–3798. doi: <https://doi.org/10.1109/ICPR48806.2021.9413340>.
- [3] K. Tong and Y. Wu, "Deep learning-based detection from the perspective of small or tiny objects: A survey," *Image Vis Comput*, vol. 123, no. 1, p. 104471, 2022, doi: <https://doi.org/10.1016/j.imavis.2022.104471>.
- [4] L. W. Sommer, T. Schuchert, and J. Beyerer, "Fast deep vehicle detection in aerial images," in *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*, Institute of Electrical and Electronics Engineers Inc., May 2017, pp. 311–319. doi: <https://doi.org/10.1109/WACV.2017.41>.
- [5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969. doi: <https://doi.org/10.1109/ICCV.2017.322>.
- [6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Adv Neural Inf Process Syst*, vol. 28, no. 6, 2015, doi: <https://doi.org/10.1109/TPAMI.2016.2577031>.
- [7] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448. doi: <https://doi.org/10.1109/ICCV.2015.169>.
- [8] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A Review of Yolo algorithm developments," *Procedia Comput Sci*, vol. 199, no. 1, pp. 1066–1073, 2022, doi: <https://doi.org/10.1016/j.procs.2022.01.135>.
- [9] X. Lu, X. Kang, S. Nishide, and F. Ren, "Object detection based on SSD-ResNet," in *2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, IEEE, 2019, pp. 89–92. doi: <https://doi.org/10.1109/CCIS48116.2019.9073753>.
- [10] V. Pandey, K. Anand, A. Kalra, A. Gupta, P. P. Roy, and B. G. Kim, "Enhancing object detection in aerial images," *Mathematical Biosciences and Engineering*, vol. 19, no. 8, pp. 7920–7932, 2022, doi: <https://doi.org/10.3934/mbe.2022370>.
- [11] H. Zhang, F. Shao, X. He, Z. Zhang, Y. Cai, and S. Bi, "Research on Object Detection and Recognition Method for UAV Aerial Images Based on Improved YOLOv5," *Drones*, vol. 7, no. 6, p. 402, 2023, doi: <https://doi.org/10.3390/drones7060402>.
- [12] M. C. Hansen *et al.*, "High-resolution global maps of 21st-century forest cover change," *Science* (1979), vol. 342, no. 6160, pp. 850–853, 2013, doi: <https://doi.org/10.1126/science.1244693>.
- [13] J. Linchant, J. Lisein, J. Semeki, P. Lejeune, and C. Vermeulen, "Are unmanned aircraft systems (UAS s) the future of wildlife monitoring? A review of accomplishments and challenges," *Mamm Rev*, vol. 45, no. 4, pp. 239–252, 2015, doi: <https://doi.org/10.1111/mam.12046>.
- [14] C. Huyck, E. Verrucci, and J. Bevington, "Remote sensing for disaster response: A rapid, image-based perspective," in *Earthquake hazard, risk, and disasters*, Elsevier, 2014, ch. 1, pp. 1–24. doi: <https://doi.org/10.1016/B978-0-12-394848-9.00001-8>.
- [15] R. Wang, Y. Murayama, and T. Morimoto, "Scenario simulation studies of urban development using remote sensing and GIS: review," *Remote Sens Appl*, vol. 22, no. 1, pp. 1–10, Apr. 2021, doi: <https://doi.org/10.1016/j.rsase.2021.100474>.
- [16] S. K. Seelan, S. Laguetta, G. M. Casady, and G. A. Seielstad, "Remote sensing applications for precision agriculture: A learning community approach," *Remote Sens Environ*, vol. 88, no. 1–2, pp. 157–169, 2003, doi: <https://doi.org/10.1016/j.rse.2003.04.007>.
- [17] E. N. Ganesh, V. Rajendran, D. Ravikumar, P. S. Kumar, G. Revathy, and P. Harivardhan, "Remote sensing analysis framework for maritime surveillance application," *International Journal of Oceans and Oceanography*, vol. 15, no. 1, pp. 11–17, 2021, Accessed: Oct. 01, 2024. [Online]. Available: [https://www.researchgate.net/publication/358549812\\_Remote\\_Sensing\\_Analysis\\_Framework\\_for\\_Maritime\\_Surveillance\\_Application](https://www.researchgate.net/publication/358549812_Remote_Sensing_Analysis_Framework_for_Maritime_Surveillance_Application)
- [18] Y. Fang, X. Guo, K. Chen, Z. Zhou, and Q. Ye, "Accurate and automated detection of surface knots on sawn timbers using YOLO-V5 model," *Bioresources*, vol. 16, no. 3, p. 5390, 2021, doi: <https://doi.org/10.15376/biores.16.3.5390-5406>.
- [19] F. Zhou, H. Deng, Q. Xu, and X. Lan, "CNTR-YOLO: Improved YOLOv5 Based on ConvNext and Transformer for Aircraft Detection in Remote Sensing Images," *Electronics (Basel)*, vol. 12, no. 12, p. 2671, 2023, doi: <https://doi.org/10.3390/electronics12122671>.
- [20] L. W. Sommer, T. Schuchert, and J. Beyerer, "Deep learning based multi-category object detection in aerial images," in *Automatic Target Recognition XXVII*, SPIE, May 2017, p. 1020209. doi: <https://doi.org/10.1117/12.2262083>.
- [21] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling, "Clustered Object Detection in Aerial Images," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 8311–8320. doi: <https://doi.org/10.1109/ICCV.2019.00840>.
- [22] J. Su, J. Liao, D. Gu, Z. Wang, and G. Cai, "Object detection in aerial images using a multiscale keypoint detection network," *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 14, no. 1, pp. 1389–1398, 2020, doi: <https://doi.org/10.1109/JSTARS.2020.3044733>.
- [23] I. Singh and G. Munjal, "Improved YoloV5 for small target detection in aerial images," *Available at SSRN 4049533*, vol. 3, no. 4049533, pp. 1–27, 2022, doi: <https://doi.org/10.2139/ssrn.4049533>.
- [24] L. Deng *et al.*, "Lightweight aerial image object detection algorithm based on improved YOLOv5s," *Sci Rep*, vol. 13, no. 1, p. 7817, 2023, doi: <https://doi.org/10.1038/s41598-023-34892-4>.
- [25] G.-S. Xia *et al.*, "DOTA: A large-scale dataset for object detection in aerial images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3974–3983. doi: <https://doi.org/10.1109/CVPR.2018.00418>.
- [26] P. Chen *et al.*, "A cascaded deep learning approach for detecting pipeline defects via pretrained YOLOv5 and ViT models based on MFL data," *Mech Syst Signal Process*, vol. 206, no. 1, pp. 110919–110935, 2024, doi: <https://doi.org/10.1016/j.ymssp.2023.110919>.
- [27] R. Huang, J. Pedoeem, and C. Chen, "YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers," in *2018 IEEE international conference on big data (big data)*, IEEE, 2018, pp. 2503–2510. doi: <https://doi.org/10.1109/BigData.2018.8621865>.
- [28] M. B. Ullah, "CPU based YOLO: A real-time object detection algorithm," in *2020 IEEE Region 10 Symposium (TENSYP)*, IEEE, 2020, pp. 552–555. doi: <https://doi.org/10.1109/TENSYP50017.2020.9230778>.
- [29] T. Jiang, C. Li, M. Yang, and Z. Wang, "An improved YOLOv5s algorithm for object detection with an attention mechanism," *Electronics (Basel)*, vol. 11, no. 16, pp. 2494–2505, 2022, doi: <https://doi.org/10.3390/electronics11162494>.
- [30] J. Zhang, Z. Chen, G. Yan, Y. Wang, and B. Hu, "Faster and Lightweight: An Improved YOLOv5 Object Detector for Remote Sensing Images," *Remote Sens (Basel)*, vol. 15, no. 20, pp. 4974–5001, 2023, doi: <https://doi.org/10.3390/rs15204974>.
- [31] M. Qiu, L. Huang, and B.-H. Tang, "ASFF-YOLOv5: Multielement detection method for road traffic in UAV images based on multiscale feature fusion," *Remote Sens (Basel)*, vol. 14, no. 14, pp. 3498–3517, 2022, doi: <https://doi.org/10.3390/rs14143498>.