Improved Object Detection in UAV Images using Deep Learning

Grishma Poudel

Nepal College of Information Technology, Nepal

grishpdl59@gmail.com

DOI: 10.47760/cognizance.2024.v04i07.009

Abstract: The use of unmanned aerial vehicles (UAV) for computer vision analysis is a significant trend in the current scenario. UAV technology is highly utilized for various purposes, including object detection, tracking, traffic management, environment monitoring, and agriculture sector, mainly due to the ease of data collection compared to conventional remote sensing methods through satellites. This study focuses on enhancing the YOLOv5 architecture to effectively detect small targets. The modifications made to the YOLOv5 framework specifically target the architecture, resulting in improved performance in object identification. The addition of a new feature fusion layer within the feature pyramid section of YOLOv5 plays a crucial role in achieving these improvements. To maintain resolution and prevent the loss of valuable feature information in the deeper sections of the network, a lateral connection is introduced, connecting this layer to an earlier part of the network. This addition ensures that crucial details and feature data are preserved throughout the network architecture. Additionally, data augmentation techniques such as image saturation and cropping are employed.

Keywords: Unmanned aerial vehicles, object detection, deep leaning, YOLOv5, Feature Fusion, augmentation.

1. INTRODUCTION

Unmanned aerial vehicle based remote sensing system is an embedded system including components as global location (GPS) and inertial measurement unit (IMU). Mainly it is used for the low altitude high resolution images. Object detection in high resolution images becomes more significant in terms of, search and rescue, intelligent agriculture, smart cities, mapping, traffic monitoring etc. Due to their excellent suitability for identifying a particular class of objects, a number of recognition application can be developed including earthquake created cracked hole detection [1], motion analysis of vehicles [2], wildlife detection and analysis [3] and many more sectors. In the case of object detection using deep neural networks there are two main challenging factors: (1) there is non-linear distribution of objects in the captured images and (2) shape, orientation, scale and other object-specific features can be differing significantly. In the case of aerial image partitioning an aerial image into numerous uniformly tiny chips and doing detection on each of them is a simple way to address the scale issue [4]. Robotics is an active area of research that involves the application of computer vision (CV) and machine learning (ML), and the use of drones as robotic platforms is becoming increasingly common.
1.1 Background of Study

The detection of small objects in images is a valuable technique, especially when dealing with aerial imagery datasets. This is because the objects within such datasets can appear very small in comparison to the overall size of the image, particularly if the image was captured from a great height. Object detection and image analysis can also be useful in the Himalayan areas for example missing object and missing person can be found out so that help can be provided in time.

To safeguard wildlife, it is crucial to set up a system for routinely monitoring animal populations, particularly during times of heightened pressure. The emergence of drones or unmanned aircraft systems (UASs) presents new possibilities. UASs offer various benefits, such as producing high-quality data at frequent intervals, collecting consistent and permanent data, having affordable operational costs, and posing minimal danger to operators. Nevertheless, UASs have certain limitations, such as having limited flight times.

Deep learning techniques are widely recognized as a prominent method for computer vision tasks such as comprehending, identifying, and categorizing images. Various techniques and studies have already been performed to detect objects in UAV-based images, such as RCNN and YOLO. There are also different versions of the YOLO algorithm. As per Nepal and Eslamiat's study in 2022 [5], the key distinctions among the architectures of YOLOv1, YOLOv2, YOLOv3, YOLOv4, and YOLOv5 are outlined. YOLOv1 employs the SoftMax function, while YOLOv2 incorporates a higher-resolution classifier, increased accuracy, and greater efficiency compared to YOLOv1. This enhancement is attributed to YOLOv2's integration of a batch normalization layer within its convolutional neural network (CNN). YOLOv3 adopts Darknet53 as its primary backbone for feature extraction from input images, yielding improved efficiency and detection performance. Notably, YOLOv3 introduces multi-object classification, allowing objects to belong to multiple categories concurrently. The SoftMax function in YOLOv3 is replaced with an independent logistic function, calculating the probability of the input image belonging to a specific label. Furthermore, YOLOv3 employs a 2-class entropy loss per category, reducing the computational complexity introduced by SoftMax functions. YOLOv4's architecture employs CSPDarknet53, a fusion of Darknet53 and CSP network, resulting in heightened accuracy, superior object detection efficiency, and reduced hardware demands. YOLOv5 introduces the Focus structure alongside CSPDarknet53 as its backbone. The novel Focus layer substitutes the initial three layers in the YOLOv3 algorithm, offering advantages such as diminished required Compute Unified Device Architecture (CUDA) memory, reduced layer count, and augmented forward and backward propagation. YOLOv5 stands out for its exceptional speed and substantial reduction in size, nearly 90% lighter than YOLOv4.

1.2 Research Objectives

The objective can be fulfilled by:

- To enhance the performance of the YOLOv5 algorithm by detecting small-sized objects presence in the UAV images.

1.3 Significance of Study

The study aims to push the boundaries of object detection technology by refining the YOLOv5 algorithm specifically for UAV imagery, enhancing its performance in complex real-world scenarios. Accurate object detection is pivotal for UAVs to effectively carry out their missions, such as identifying and tracking objects of...
interest, avoiding obstacles, and responding to dynamic environmental changes. By improving the object detection capabilities of YOLOv5, this study aims to enhance the operational effectiveness and reliability of UAVs in diverse environments.

2. LITERATURE REVIEW

Recent advances in deep learning have led to significant improvements in multi-class picture identification. Successful approaches for picture identification are presented using the ImageNet [6] and Pascal VOC [7] challenges, utilizing convolutional neural network (CNN) blocks combined with other features such as max-pooling. An extremely deep CNN is suggested for image recognition in [8] in 2012, and this leads to a significant improvement in picture recognition. Inception modules for convolutional blocks and skip connections were introduced in 2014 [9]. The performance of an extremely deep network with skip connections has been excellent. Deconvolutional single shot detector introduces ResNet101 to SSD and uses deconvolution layers to provide a better detection approach that can work better with small-scale objects [10]. To address the class imbalance issue in object recognition, focal loss for dense object detection [11] introduces a novel loss termed Focal Loss. A module that improves anchor boxes is then suggested in the paper on “single shot refinement neural network for object detection” [12]. Where they suggested a brand-new single-shot-based detector dubbed RefineDet that achieves better accuracy than two-stage methods and maintains comparable efficiency of one-stage methods. In Detecting objects as paired key points [13], corner pooling is added, objects are recognized as two key points defining the bounding box, and anchor boxes are completely abolished.

Due to the recent emergence of drone imagery and the lack of publicly available datasets until a few years ago, there is a shortage of literature on object detection in drone images. The VisDrone dataset, which was first released in 2018, [14] is one of the largest drone datasets available for object detection. During the VisDrone challenges, it was observed that various adaptations of YOLOv3 algorithm were used in multiple solutions. One such solution involved the fusion of YOLOv3 and Faster R-CNN algorithms [14]. A proposed car detection algorithm for aerial images incorporated both YOLOv3 and R-CNN techniques. The results reported in the paper indicated that YOLOv3 was more effective in detecting cars in aerial images compared to R-CNN [15]. Darknet53 is the foundational architecture of YOLOv3 [16]. It comprises of 53 convolution layers and 23 residual layers, and does not include pooling layers in its structure. Instead, the transformation of tensor dimensions is achieved by adjusting the stride of convolution kernel. For instance, setting stride = (2,2) reduces the edge resolution by half [16]. Although YOLOv3 has shown good performance on the COCO dataset, its results were not satisfactory when used directly to detect vehicles in aerial images. The recall and precision rates were only 84.3% and 86.7% respectively [17].

2.1 YOLO Version 5

YOLO v5 is an improved version of the YOLO algorithm developed by the Ultralytics team [18]. In addition, Glenn Jocher, the creator of the Mosaic data augmentation technique, has been recognized by Alexey Bochkovsky in the YOLOv4 paper [19]. However, his YOLOv5 model sparked significant controversy within the computer vision community due to both its name and the advancements it introduced. Nonetheless, YOLOv5 offered notable engineering benefits. Unlike its predecessors, YOLOv5 was implemented in the
Python programming language instead of C. This change facilitated simpler installation and integration on Internet of Things (IoT) devices. Furthermore, the PyTorch community, which YOLOv5 is based on, is larger than the Darknet community associated with previous versions. This implies that PyTorch is likely to receive more contributions and has greater growth potential in the future. However, comparing the performance of YOLOv4 and YOLOv5 accurately is challenging due to their utilization of different programming languages and frameworks. Over time, YOLOv5 demonstrated superior performance compared to YOLOv4 in specific scenarios, which contributed to gaining some confidence within the computer vision community, alongside YOLOv4.

As mentioned previously, the YOLOv5 architecture incorporates the latest advancements in a similar manner to YOLOv4, resulting in minimal notable differences in theory. The author opted not to publish a comprehensive paper but instead created a repository on GitHub where they continuously update and improve the model. By analyzing the code structure in the .yaml file, the YOLOv5 model can be summarized as follows.

- **Backbone:** Utilizes the Focus structure and CSP network.
- **Neck:** Includes the SPP block and PANet.
- **Head:** Adopts the YOLOv3 head with the GIoU-loss.

The YOLOv5 author emphasizes a notable engineering distinction, which relates to Joseph Redmon's introduction of the anchor box structure in YOLOv2 [20]. Redmon proposed a method [20] for selecting anchor boxes that closely match the size and shape of the ground truth bounding boxes in the training set. Using the k-means clustering algorithm with varying values of $k$, the YOLOv5 authors selected the top 5 anchor boxes that best fit the COCO dataset, which comprises 80 classes [18]. These chosen anchor boxes are set as the default, resulting in reduced training time and improved network accuracy.

### 3. METHODOLOGY

The YOLO algorithm was employed in this study for the detection of aerial images based on the VisDrone dataset. The efficiency of object detection varies as the height of the drone and angle of the camera change. The goal of data augmentation is to make the input images more variable so that the developed object detection model is more resilient to images taken in various contexts. In this study, photometric and geometric distortion augmentation techniques were used, which positively contribute to the task of object recognition. To enhance the detection of small targets in aerial images, a feature fusion layer has been incorporated into the feature pyramid network of YOLOv5. Also, to enhance the detection of small targets in aerial images, a feature fusion layer has been incorporated into the feature pyramid network of YOLOv5. Since the dataset includes objects that are extremely small, consisting of only a few pixels in width, the suggestion is to introduce an additional output fusion layer with a lower stride to effectively detect these tiny objects within the image. A lower stride will result in smaller grid cells, allowing the model to focus its attention on small areas of the image individually. Furthermore, the output layer is combined with earlier layers in the feature pyramid through concatenation. By incorporating the new blocks from the beginning of the network, the model's ability to identify and handle small objects will be preserved. YOLOv5's head section includes different output levels that
correspond to detections at various scales. However, since the dataset contains extremely small objects that are only a few pixels wide, a proposal is made to introduce an additional output fusion layer with a lower stride. This layer aims to specifically detect these tiny objects within the image. By utilizing a lower stride, the resulting grid cells will be smaller, which in turn directs the model’s attention towards smaller areas of the image. Furthermore, the output layer is concatenated with earlier layers in the feature pyramid to maintain the model’s ability to handle small objects effectively.

**Figure 1: Block Diagram of methodology**

The figure below illustrates the comprehensive system workflow diagram of this research, in which the performance of both the default and proposed models is assessed until an improvement in object detection is achieved. The proposed modifications in the YOLOv5 architecture aim to enhance object detection accuracy, especially in scenarios involving small and challenging objects within images. By customizing the backbone to transmit additional spatial information, the model becomes more adept at capturing intricate details.

**Figure 2: Workflow diagram**
3.1 Data Augmentation

Data Augmentation plays a crucial role in improving object detection performance in aerial images using YOLOv5. By leveraging data augmentation techniques, the dataset can be enriched with diverse variations of the original images, thereby enhancing the model's ability to generalize and detect objects accurately. Augmentation methods such as random cropping, rotation, scaling, flipping, and color jittering effectively introduce variations that simulate real-world scenarios. These techniques not only increase the size of the training dataset but also improve the model's robustness and adaptability to unseen aerial images. Furthermore, data augmentation helps mitigate the problem of imbalanced class distribution by creating additional samples for underrepresented classes. Through careful application of data augmentation strategies, the performance of YOLOv5 can be significantly enhanced, leading to improved object detection accuracy in aerial images.

In this study, the image saturation technique was applied to a set of images captured in low light conditions, resulting in a positive impact on the detection mechanism, with saturation levels increased up to 10 percent. Additionally, for images taken from very high heights and at irregular angles, cropping ranging from 0 to 5 percent was performed. In general, highly saturated colors tend to provide more distinct and distinguishable visual features, which can aid in object detection. When the colors are vibrant and saturated, the edges and boundaries of objects in an image are more pronounced, making it easier for the detection mechanism to identify and differentiate objects from the background. When colors are desaturated, the visual contrast between objects and the background decreases, making it more difficult for the detection mechanism to accurately distinguish and localize objects. Also, cropping in an image dataset plays a significant role in enhancing detection accuracy by focusing on relevant regions of interest and removing irrelevant or distracting information. In the dataset containing 10 classes, the augmentation primarily focuses on the underrepresented classes found in the VisDrone datasets. For better dataset balancing, images containing objects that are underrepresented in the entire dataset are augmented. A total of 330 images in the dataset were augmented, resulting in the addition of 660 images to the entire dataset. The sample of augmented images is shown in the figure below.

Figure 3: Sample of augmented images
3.2 Proposed YOLOv5 Architecture

By employing the concept of compound scaling, the width scaling is reduced by a factor of 0.8 while the depth scaling remains unchanged at 1x. The resultant model has achieved significantly improved accuracy while experiencing only a slight increase in size. In default YOLOv5 architecture utilizes different output levels in its head section to identify objects at various scales. There are three anchor boxes are employed per scale, resulting in a total of nine anchor boxes in the standard YOLOv5 configuration.

Figure 4: Default YOLOv5 Architecture

In the above architecture, the output section contains outputs from different strides: 8, 16, and 32. To address the presence of tiny objects in the dataset, a suggestion is made to incorporate an additional output fusion layer with a lower stride. This layer aims to detect extremely small objects within the image. By using a lower stride, the resulting grid cells become smaller, directing the model's focus to smaller regions of the image. Additionally, the output layer is combined with earlier layers in the feature pyramid. This concatenation of new blocks from the start of the network helps maintain high-resolution feature maps, enabling the detection of small objects in the image. The proposed model in this study builds upon the YOLOv5 P6 model shown in Figure 5 below, by
introducing an additional output layer. This new layer is inserted into the YOLOv5 regular model with a stride of 64. To further enhance small target detection, the proposed model takes another step forward by including an additional output layer at the beginning of the pyramid with a stride of 4, as depicted in Figure 6. In this context, the term "Conv" refers to a regular Convolutional block, "C3" represents a BottleneckCSP block comprising three convolutions, "SPP" denotes Spatial Pyramid Pooling, "Concat" signifies the concatenation function in PyTorch, and "Upsample" is an Upsample function used to enlarge the image using K nearest neighbors.

Figure 5: Default YOLOv5P6 Architecture
In the context of machine learning, loss functions serve as a guide for training algorithms to adjust the parameters of a model in a way that minimizes the error between predictions and ground truth. The goal is to find the optimal parameters that make the model's predictions as close as possible to the true values. The total loss is calculated as:

$$Loss = L_{\text{class}} + L_{\text{obj}} + L_{\text{box}}$$  \hspace{1cm} (i)$$

Furthermore, $L_{\text{box}}, L_{\text{cls}}$ and $L_{\text{obj}}$ represent the loss functions for bounding box regression, classification and confidence respectively. The expression defining bounding box regression is as follows:

$$L_{\text{box}} = \frac{1}{2} \sum_{i=0}^{w} \sum_{j=0}^{h} \left( (x_i - \bar{x})^2 + (y_i - \bar{y})^2 + (w_i - \bar{w})^2 + (h_i - \bar{h})^2 \right)$$  \hspace{1cm} (ii)$$

The classification loss is expressed as:

$$L_{\text{cls}} = \sum_{i=0}^{w} \sum_{j=0}^{h} p_i \log \tilde{p}(c)$$  \hspace{1cm} (iii)$$

The Confidence loss is written as:

$$L_{\text{obj}} = \lambda \sum_{i=0}^{w} \sum_{j=0}^{h} \left( (C_i - \bar{C})^2 + \lambda \sum_{i=0}^{w} \sum_{j=0}^{h} \left( (C_i - \bar{C})^2 \right) \right)$$  \hspace{1cm} (iv)$$

Where:

- $x_i, y_i, w_i, h_i$ are the predicted coordinates and dimensions of the bounding box in the i-th grid cell and j-th bounding box.
- $\bar{x}_i, \bar{y}_i, \bar{w}_i, \bar{h}_i$ are the corresponding ground truth values.
• $C_i$ represents the confidence score for the presence of an object in the i-th grid cell and j-th bounding box.
• $\overline{c}_i$ is the ground truth confidence score.
• $P_i(c)$ is the predicted probability of class c in the i-th grid cell.
• $\overline{P}_i(c)$ is the corresponding ground truth probability.
• $1^{	ext{obj}}_{ij}$ is an indicator function that is 1 if object j is assigned to cell i, and 0 otherwise.
• $\lambda_c$ and $\lambda_{	ext{obj}}$ are hyperparameters that control the influence of the coordinate and confidence terms, respectively.

### 3.3 Datasets

#### 3.3.1 VisDrone

The VisDrone dataset [14] for aerial object detection consists of more than 6,000 aerial images taken by camera-equipped unmanned air vehicles. In total, there are ten classes that are used in the evaluation in both [14] and this work, which may be listed as: 0. pedestrian, 1. people, 2. bicycle, 3. car, 4. van, 5. truck, 6. tricycle, 7. awning-tricycle, 8. bus, and 9. motor.

![Pie chart describing the number of instances of labels for each class](image)

The VisDrone dataset provides detailed annotations for object detection and tracking tasks. Each image or frame in the dataset is labeled with bounding boxes around the objects of interest, along with associated class labels. The VisDrone dataset is divided into different subsets for training validation and test. The k-fold cross-validation technique, which will be explained in detail in the result and analysis section, was applied for the optimized division of the dataset into train and validation folders.

#### 3.3.2 Dataset Preparation

The VisDrone dataset annotations were also transformed into YOLO format. As stated in the VisDrone documentation [14], the VisDrone format represents bounding box annotations as follows: bbox left, bbox top, bbox width, bbox height, score, object category, truncation, occlusion. In this format, “bbox left” denotes the x-coordinate of the top-left corner of the predicted bounding box, "bbox top" represents the y-coordinate of the top-left corner of the predicted object bounding box, "bbox width" indicates the width of the predicted object bounding box in pixels, and "bbox height" corresponds to the height of the predicted object bounding box in pixels.
pixels. Following figure illustrates the process of converting VisDrone format annotations into YOLO format annotations.

![VisDrone to YOLO conversion diagram](image)

Figure 8: VisDrone Annotations in YOLO Format

The VisDrone format conveniently provides the necessary information of the top-left coordinates (x and y) of the bounding box, as well as the width and height, which can be directly utilized. VisDrone annotation format to YOLO annotation format conversion.

and, y coordinate will be:

\[
x = \text{width} + \frac{x}{2}
\]

\[
y = \text{height} + \frac{y}{2}
\]

![Sample images from the VisDrone datasets](image)

Figure 9: Sample of images from the VisDrone datasets
3.4 Accuracy Metric

To determine accuracy, the evaluation employs the mean average precision (mAP) metric. Both mAP at IoU (Intersection over Union) threshold of 0.5 and mAP ranging from 0.5 to 0.95 are recorded and examined. While precision and recall are commonly employed metrics in various classification tasks, object detection necessitates a more comprehensive metric that can provide insights into the complexities involved in detecting and precisely locating an object within an image. Precision and recall are calculated as follows:

\[
\text{Precision} = \frac{TP}{TP + FP} \\
\text{Recall} = \frac{TP}{TP + FN} \\
mAP = \frac{\sum_{i=0}^{N-1} \left( \int_{R_i} p(R) \, dR \right)}{N}
\]

In this context, "TP" represents True Positives, "FP" represents False Positives, and "FN" represents False Negatives. In the case of object detection, a detection is deemed a true positive when the bounding box of the prediction overlaps with the bounding box of the ground truth by a specific threshold value. The evaluation of network performance involves three metrics: P (precision), R (recall), and mAP (mean average precision).
3.5 Tools and Resources

3.5.1 Google Colaboratory

The cloud-based platform provided by Google, commonly known as Google Colab, offers an interactive computing environment through Jupyter notebooks. It eliminates the need for local installations or setups since it is entirely run in the cloud, accessible through a web browser. The free version of Google Colab was utilized for the analysis of the proposed model. In this version, a total RAM capacity of 12.7 GB was employed. Additionally, the GPU (Graphics Processing Unit) RAM had a total capacity of 15.0 GB. The disk space allocated for the analysis had a total capacity of 78.2 GB. Different numbers of epochs were run in Colab during the analysis of the proposed model.

3.5.2 LabelImg

LabelImg is an open-source graphical image annotation tool that allows to draw bounding boxes around objects and assign corresponding labels. It supports various annotation formats, including YOLO format, making it compatible with YOLO training.

3.5.3 Keras ImageDataGenerator:

Keras, a popular deep learning library, includes the ImageDataGenerator class, which provides built-in image augmentation functionalities.

4. RESULTS AND DISCUSSION

The image dataset is uploaded to Google Colab, where the path is easily set to retrieve the data. Initially, the dataset is augmented using the Keras ImageGenerator. The images are saturated by up to 10%, and those captured from high altitudes undergo a 5% cropping process. During cropping, objects of any category are carefully noted and not eliminated from the images. The proposed model is constructed using the official YOLOv5 GitHub repository. Modifications are made to a custom model yaml file, and training is conducted on the Google Colab platform using Jupyter notebooks. The YOLOv5 developers have simplified the configuration of the repository for creating custom models by creating a yaml file to define the architecture.

4.1 Dataset Division

At first, dataset division and check were employed to divide the datasets into train and valid folders. Ratios of 70:30, 80:20, and 90:10 were used to partition the entire dataset. The performance of the YOLOv5 model was assessed for different epochs in Colab to analyze whether the dataset division resulted in underfitting, overfitting, or a good fit. Initially, for the 70:30 ratio, the images in the dataset were divided into the train and valid folders, comprising 4991 and 2139 images, respectively. A cross-validation approach was used to maintain the correct balance between the number of images in the train and valid folders. This approach made use of a Roboflow API tool that also examined the health of the images in the datasets. The performance of the YOLOv5 model was evaluated after dividing the datasets in the 70:30 ratio, and the figure below shows that this division was not ideal.
Similarly, an 80:20 ratio was applied to the datasets to figure out the optimality of the dataset divide. The 7130-image dataset has been divided into 5 files, each holding data from 1426 images. This solution made use of a Roboflow API tool, which also looked at the health of the images in the datasets. Following that, the datasets were separated into two folders, with 80% (5704 images) in the train folder and 20% (1426 images) in the validation folder. The performance of the YOLOv5 model was evaluated after splitting the datasets in an 80:20 ratio. The figure below shows that, unlike in the past, this divide was effective.

Figure 11: Results of 70:30 Dataset Division

Figure 12: Results of 80:20 Dataset Division
After that, the datasets were subjected to a ratio of 90:10 to assess the optimality of the dataset division. The performance of the YOLOv5 model was evaluated. The following figure illustrates that this division is also not satisfactory.

![Figure 13: Results of 90:10 Dataset Division](image)

In this manner, confirmation was attained that the optimal dataset division corresponds to an 80:20 ratio. After the division of datasets into an 80:20 ratio, the class-wise distribution of VisDrone datasets is explained in the table below (Table 1).

<table>
<thead>
<tr>
<th>Class</th>
<th>All</th>
<th>Train</th>
<th>Val</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pedestrian</td>
<td>79337</td>
<td>63090</td>
<td>16247</td>
</tr>
<tr>
<td>People</td>
<td>27059</td>
<td>20742</td>
<td>6317</td>
</tr>
<tr>
<td>Bicycle</td>
<td>10480</td>
<td>7143</td>
<td>3337</td>
</tr>
<tr>
<td>Car</td>
<td>144867</td>
<td>114466</td>
<td>30401</td>
</tr>
<tr>
<td>Van</td>
<td>24956</td>
<td>19933</td>
<td>5023</td>
</tr>
<tr>
<td>Truck</td>
<td>12875</td>
<td>10613</td>
<td>2262</td>
</tr>
<tr>
<td>Tricycle</td>
<td>4812</td>
<td>4292</td>
<td>520</td>
</tr>
<tr>
<td>Awning-tricycle</td>
<td>3246</td>
<td>2775</td>
<td>471</td>
</tr>
<tr>
<td>Bus</td>
<td>5929</td>
<td>4543</td>
<td>1386</td>
</tr>
<tr>
<td>Motor</td>
<td>29647</td>
<td>23920</td>
<td>5727</td>
</tr>
</tbody>
</table>

### 4.2 Performance Evaluation

For accurate calculation and comprehensive analysis of the model, the training phase is divided into two parts. In the first part, training is conducted for comparison between the default YOLOv5 (passing strides of 8, 16, and 32) and the default YOLOv5p6 (passing strides of 8, 16, 32, and 64) models for 60 epochs with 32 batches. It shows that the YOLOv5p6 model performs better as compared to the default YOLOv5 model. Then all the comparisons are taken on the basis of the YOLOv5p6 model, and all parameters are documented in Table 2. In
the second part, the proposed YOLOv5 model is trained for 60 epochs, and all result outcomes are also recorded in Table 2. To ensure a fair comparison, all models were trained from scratch without utilizing any pretrained weights for an equal number of epochs. The metric employed for accuracy measurement and performance assessment is mean average precision (mAP), as previously mentioned. The results are presented in terms of mAP at 0.5 intersection over union (IoU), as well as mAP ranging from 0.5 to 0.95, on the validation datasets for VisDrone. The mean average precision (mAP@0.5) is determined using a single IoU (intersection over union) criterion of 0.5.

The overlap between a predicted bounding box and a ground-truth bounding box is measured by IoU. An IoU of 0.5 indicates that the overlap between the predicted box and the ground truth box must be at least 50% for a detection to be judged accurate and mAP@0.5 measures how effectively the model detects objects when it has at least a modest overlap with the ground truth. mAP@0.5:0.95 also refers to the mean average precision determined over a range of IoU thresholds, namely from 0.5 to 0.95. This mAP variation takes numerous degrees of overlap between predicted and ground truth bounding boxes into account, giving a more complete evaluation of the model's performance across different levels of object localization accuracy. mAP@0.5 evaluates the model's performance across varied amounts of object overlap by focusing on a single IoU threshold of 0.5, whereas mAP@0.5:0.95 analyzes a larger range of IoU thresholds.

<table>
<thead>
<tr>
<th>Train Models</th>
<th>mAP@0.5</th>
<th>mAP@0.5:0.95</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default YOLOv5 model.</td>
<td>0.317</td>
<td>0.157</td>
</tr>
<tr>
<td>Default YOLOv5p6 model.</td>
<td>0.380</td>
<td>0.201</td>
</tr>
<tr>
<td>Proposed YOLOv5 model.</td>
<td>0.410</td>
<td>0.210</td>
</tr>
</tbody>
</table>

The findings suggest that the suggested enhancement to the YOLOv5 model led to an improvement in detection accuracy. Specifically, an increase of 7.89% was observed in mean Average Precision at IoU=0.5 (mAP@0.5), along with a 4.48% increase in mean Average Precision at mAP@0.5:0.95. The figures visually demonstrate the effectiveness of the proposed model in addressing the issue of object detection misses, when compared to the default version of YOLOv5p6.
In order to evaluate the performance of the proposed YOLOv5 model, a representative small object image from the VisDrone dataset was chosen for visual comparison. The detection effects of both the original model (shown in Figure 14) and the proposed YOLOv5 model (shown in Figure 15) were observed. Differences between the images were indicated using indication circular boxes. Upon comparing the two, it becomes evident that the
proposed YOLOv5 model addresses the issue of missed detections that existed in the default YOLOv5p6 model, thus demonstrating an improvement. The selected small objects in image from the VisDrone test set served as a representative sample to assess the detection performance of the proposed YOLOv5 algorithm model. By visually comparing the detection results showcased in above figures, it becomes apparent that there are distinct disparities between the original model and the proposed YOLOv5 model (Figure 14 and Figure 15, respectively). These discrepancies are visually highlighted using circular boxes, indicating areas where the proposed model outperforms the default YOLOv5p6 model by mitigating missed detection issues. Furthermore, Figure 16 illustrates the object detection capabilities of the proposed YOLOv5 model, showcasing identified objects along with their corresponding labels.

The provided table (Table 3) displays the accuracy metrics mAP@0.5 and mAP@0.5:0.95 for different classes on the VisDrone dataset. It compares the default YOLOv5p6 model with the proposed YOLOv5 model. The enhanced model demonstrates a notable improvement in both mAP@0.5 and mAP@0.5:0.95 metrics across most of the classes. Particularly, when it comes to classes containing small objects like small vehicles and large
vehicles, the enhanced model outperforms the regular YOLOv5p6 model significantly. The enhanced model demonstrates significant improvement, particularly when dealing with classes that involve small objects such as small vehicles and large vehicles. In these specific cases, the enhanced model surpasses the regular YOLOv5p6 model by a substantial margin. These results serve as compelling evidence for the effectiveness of the proposed enhancements in accurately detecting and classifying objects.

Table 3: Class wise mAP@0.5 and mAP@0.5:0.95 of Default YOLOv5p6 and Proposed YOLOv5

<table>
<thead>
<tr>
<th>Class</th>
<th>mAP@0.5 Default YOLOv5p6</th>
<th>mAP@0.5 Proposed YOLOv5</th>
<th>mAP@0.5:0.95 Default YOLOv5p6</th>
<th>mAP@0.5:0.95 Proposed YOLOv5</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>0.380</td>
<td>0.410</td>
<td>0.201</td>
<td>0.210</td>
</tr>
<tr>
<td>Pedestrian</td>
<td>0.349</td>
<td>0.380</td>
<td>0.137</td>
<td>0.141</td>
</tr>
<tr>
<td>People</td>
<td>0.226</td>
<td>0.226</td>
<td>0.0703</td>
<td>0.0798</td>
</tr>
<tr>
<td>Bicycle</td>
<td>0.133</td>
<td>0.134</td>
<td>0.0438</td>
<td>0.0508</td>
</tr>
<tr>
<td>Car</td>
<td>0.698</td>
<td>0.728</td>
<td>0.431</td>
<td>0.489</td>
</tr>
<tr>
<td>Van</td>
<td>0.472</td>
<td>0.482</td>
<td>0.304</td>
<td>0.305</td>
</tr>
<tr>
<td>Truck</td>
<td>0.515</td>
<td>0.525</td>
<td>0.298</td>
<td>0.299</td>
</tr>
<tr>
<td>Tricycle</td>
<td>0.237</td>
<td>0.247</td>
<td>0.107</td>
<td>0.110</td>
</tr>
<tr>
<td>Awning-tricycle</td>
<td>0.168</td>
<td>0.188</td>
<td>0.0909</td>
<td>0.0951</td>
</tr>
<tr>
<td>Bus</td>
<td>0.653</td>
<td>0.673</td>
<td>0.409</td>
<td>0.415</td>
</tr>
<tr>
<td>Motor</td>
<td>0.339</td>
<td>0.439</td>
<td>0.119</td>
<td>0.123</td>
</tr>
</tbody>
</table>
The results of the default YOLOv5p6 model and the Proposed model, trained on the VisDrone dataset, are depicted in Figure 17 and 18. These figures showcase various metrics, including the bounding box losses, the average value of target detection loss, and the average value of classification loss on both the training and validation dataset. A comparison of Figure 17 and 18 reveals that the new model not only improves the detection accuracy but also significantly enhances the recall rate.

In the results graph, box losses represent the box loss during the training and validation phases. They measure the discrepancy between the predicted bounding box coordinates and the ground truth bounding box coordinates. Upon comparing the graphs obtained from the default and proposed models, it is evident that the proposed model achieves lower values for these losses, indicating an improvement in object localization accuracy. Additionally, the object loss reflects the difference between the predicted objectness score and the ground truth objectness label during training and validation. Minimizing this loss enhances the detection of objects, as shown in Figure 18. The class loss represents the classification loss, which quantifies the dissimilarity between the predicted class probabilities and the ground truth class labels. The model strives to
minimize this loss to enhance the accuracy of object classification. Precision, a performance metric, calculates the proportion of correctly predicted positive detections (True Positives) out of all positive predictions (True Positives + False Positives). In the YOLOv5 algorithm, precision assesses object detection accuracy by evaluating the model's ability to avoid false positives. Recall, also known as sensitivity or true positive rate, evaluates the proportion of correctly predicted positive detections (True Positives) out of all actual positive instances (True Positives + False Negatives). It assesses the model's capability to avoid false negatives and also indicating its proficiency in capturing all relevant objects in the image.

The comparison of mean Average Precision (mAP) results between the default and improved YOLOv5 models offers valuable insights into the advancements achieved through the proposed enhancements. The mAP metric serves as a comprehensive measure of overall detection performance, considering precision and recall at a specific intersection over union (IoU) threshold. The figures presented above clearly demonstrate that the proposed model outperforms the default model in terms of object localization and classification.

A confusion matrix is a matrix that describes classification accuracy performance of a model. It is used in identifying object tasks to measure the model's capacity to appropriately categorize and find things. A confusion matrix is a square matrix with rows and columns representing the different classes or categories for which the model has been trained. Each cell in the matrix represents the number of predictions generated by the model for
a certain class and how well they match the ground truth labels. The confusion matrix is divided into four primary categories:

- **True Positives (TP):** This cell indicates the number of correct positive predictions made by the model.
- **False Positives (FP):** This cell represents the number of incorrect positive predictions made by the model.
- **False Negatives (FN):** This cell indicates the number of incorrect negative predictions made by the model.
- **True Negatives (TN):** This cell represents the number of correct negative predictions made by the model.

By examining the values in the confusion matrix, as shown in the figures below, it becomes evident that the proposed model possesses a superior ability to accurately classify and localize objects.

![Confusion Matrix](image)

Figure 19: Confusion matrix of default YOLOv5p6 model
Also, precision, recall, and the F1 curve are significant assessment measures used in object-detecting tasks. As a result, the following parts provides an explanation of the comparison between the default and proposed models based on these factors.
The F1 curve represents the harmonic mean of precision and recall. It combines both metrics to provide an overall assessment of the model’s performance. The F1 curve shows the trade-off between precision and recall by varying the confidence threshold for classifying objects. In the above F1 curves of the default and proposed models, the proposed model consistently outperforms the default model in terms of the F1 score. This superiority indicates a better balance between precision and recall, demonstrating an improvement in object detection accuracy.

The recall confidence curve in the YOLOv5 algorithm offers insights into the model’s capability to detect objects across varying confidence thresholds. It illustrates the relationship between recall (true positive rate) and the confidence threshold employed for object detection. By comparing the recall confidence curves of the default and improved models, as demonstrated in the figures below, it becomes possible to assess the effectiveness of the proposed enhancements. This comparison enables the evaluation of the improved model’s capacity to achieve higher recall rates, which in turn signifies superior detection performance.

Figure 21(a): F1-Confidence curve of Default YOLOv5p6

Figure 21(b): F1-Confidence curve of Proposed model
Figure 22 (a): Recall-Confidence Curve of default model
Figure 22(b): Recall-Confidence curve of proposed model

Figure 23 (a): Precision Recall Curve of default model
Figure 23(b): Precision Recall curve of proposed model
5. CONCLUSION AND RECOMMENDATIONS

Finally, this study presents an improved version of the YOLOv5 architecture suited for recognizing small objects in UAV based images. The new model was trained using VisDrone datasets and compared to the regular YOLOv5p6 model. The findings are outstanding, with a significant 7.89% gain in the mAP@0.5 benchmark and a 4.45% increase in the mAP@0.5:0.95. The new model achieved a mAP score of 0.410, which is greater than the mAP score of 0.380 obtained from the default YOLOv5p6 model on the VisDrone dataset. This research showed that increasing the transmission of extra spatial information from the lower stride of the model to the detection head of the model considerably improves the model's performance in recognizing small objects. Based on the findings of this study, it is suggested that researchers continue to work on improving and expanding the way by which extra spatial information is communicated from the model's bottom half to the detecting section.

Exploration of other techniques in this area might help to additional accuracy improvements.

REFERENCES