Ingénierie des Systèmes d'Information
Vol. 28, No. 2, April, 2023, pp. 491-498
Journal homepage: http://iieta.org/journals/isi

Camshift Algorithm with GOA-Neural Network for Drone Object Tracking

Lokesh Sai Kiran Vatsavai*, Krishna Satya Varma Mantena

Department of Information Technology, SRKR Engineering College, Bhimavaram 534204, India

Corresponding Author Email: 21B91D4003@srkrec.ac.in

https://doi.org/10.18280/isi.280226

Received: 10 February 2023
Accepted: 2 April 2023

Keywords:
object detection, tracking system, Camshift algorithm, neural network, AU-AIR Drones

ABSTRACT

Detecting objects in scenes filmed by drones is a trendy new activity. Since drones are constantly changing altitude, the magnitude of the objects they encounter wildly fluctuates, making it difficult to optimise networks. However, because of the complexity of the environment, such tracking approaches cannot be functional for real-world issues. For instance, the tracking system has a hard time locating people of interest when there are several of them in close proximity to one another. Another major factor in the system's inability to detect individuals is the prevalence of backgrounds of a similar hue. In order to do this, this study suggests using a Camshift method in tandem with an optimal-based neural network. When compared to methods that rely on conventional tracking algorithms, this one is both more cost-effective and flexible in different settings. This model makes adjustments to the Yolo neural network, the Camshift algorithm, and other previously merged components. The issues with occlusion, lighting, scale, and noise in the Camshift algorithm are addressed. We conducted our tests using two publicly available datasets: VisDrone and AU-AIR. Experiments using the VisDrone and AU-AIR datasets demonstrate the suggested method's ability to dramatically enhance classification accuracy.

1. INTRODUCTION

Object identification techniques have been used in many real-world applications, such as crop protection [1, 2], animal protection [3, 4], and city monitoring [5, 6]. In this study, we want to learn more about the above-mentioned multiple applications by making it easier to identify objects in photos taken by drones. Significant advances in object identification tasks employing deep convolutional neural networks [4, 5] have been made in current years. Benchmark datasets that are very important, like MS COCO and PASCALVOC, help a lot to make object detection software better. Previous deep CNN, on the other hand, are frequently optimised for images of natural scenes [6]. There are primarily three issues with using already-existing models to take on the item detection task using drone-captured settings [7]. Due to the wide range of drone altitudes, there is a dramatic shift in perceived object size. Two, there is occlusion between items because of the high density of objects in drone-captured photos. Third, because drones can photograph such a vast region, there are always geographical details that are unclear in the resulting photographs. Object recognition in drone-captured photos is notoriously difficult because of the aforementioned issues [8, 9].

To make embedded systems [9], the main goal of these designs is to improve (1) how well they find things and (2) how hard their methods are to figure out. According to its top-level architecture, CNN-based object detectors may be broken down into two groups: (1) region-based indicators, and (2) single-shot sensors [10].

A region-based detector is usually made up of a region-proposal phase and a classifier phase. An improved version of the R-CNN is a type of region-based detector. The main drawback of region-based sensors is that they are intensive, making it challenging to attain excellent presentation in embedded systems. One-shot detectors use a single CNN to carry out all phases of the object identification process. There are several types of detectors, but some of the most common are single-shot detectors like YOLO and SSD. Because of its intended use in real-time applications, YOLO offers a performance-accuracy trade-off that is inherently biassed towards the former [11].

In the last few decades, CV tasks like identifying objects and dividing up pictures have become more and more common. The recognition of certain classes of visual objects (such as automobiles, pedestrians, animals, terrains, etc.) in photographs is a difficult but valuable task known as object detection (OD) [12]. The creation of computational models and methods is the focus of OD, which is the most significant issue in vision. Segmentation, picture captioning, object tracking, etc. all build off of it as a foundational work. Therefore, OD is applicable to a wide variety of fields, including, but not limited to, face detection, pedestrian finding, distant satellite finding, etc. [13]. In this paper, we apply our innovative methodology to the problem of OD in drone photos from two datasets: the VisDrone 19 test dataset [14] and the dataset [15, 16].

Due to the lack of UAV duplicate datasets, optical distortion (OD) in drone photos presents a significant issue for computer vision, and this study explores both data increase and DL approaches for OD in drone images. The most significant contributions are as follows:

(1) We utilise the multi-frame subtraction approach to find and follow a moving target, and then we calibrate the target area using a connected region search.

(2) The proposed time can be shortened if the target region is first determined.

(3) We present a new metaheuristic optimization technique called GOA that uses a swarm intelligence approach to fine-tune the parameters of deep learning networks in a manner reminiscent to the predatory behaviour of gannets.

(4) The U- and V-shaped headfirst designs of Gannets are the inspiration for GOA's exploration phase, while the rapid revolution and random walk of its expansion phase guarantee that a superior solution will be discovered in the region.

Here's how the breather of the paper is placed out: The relevant literature is presented in Section 2, and the suggested model is described in Section 3. In Section 4, we present an assessment of the projected model using standard methods. In the final section, we discuss how this study will inform further investigations.

2. RELATED WORKS

Traditional approaches for diagnosing plant diseases have limited efficacy due to characteristics such as dense circulation, uneven, multi-scale object classes, and textural similarities. Roy et al. [17] describe a high-presentation real-time fine-grain object documentation system to address these challenges. The suggested model is based on You Only Look Once (YOLOv4), an improved version of the original procedure. A redesigned Path Aggregation Network (PANet) keeps fine-grain localised info and improves feature fusion; spatial pyramid pooling (SPP) enlarges the receptive arena; and DenseNet is employed in the backbone to optimise reuse. The projected model has an F1-score detection rate of 93.64% and a mAP value of 96.29%. The existing body of work provides an efficient and effective approach for identifying numerous plant illnesses in complicated settings, with possible applications extending to the detection of a variety of fruits and crops, as well as general disease detection and automated agricultural detection procedures.

To improve the efficiency of the foundational models used for multiscale thing recognition in drone images, Walambe et al. [18] adopt ensemble transfer learning (ETL). In order to recognise objects of varying sizes in UAV photos, the system utilises a test-time augmentation pipeline that mixes many models and employs voting mechanisms. Additionally, the data augmentation provides an answer to the problem of insufficient drone picture datasets. We conducted our tests using two publicly available datasets: VisDrone and AU-AIR. Instead of spending time and resources training unique models on complete datasets, we employ transfer learning and a two-level voting technique collaborative to get better results. Employing ensemble transfer learning results in a notable increase in mAP on both the VisDrone and AU-AIR datasets, as demonstrated experimentally. Additionally, the end-user may pick and track the impacts of the method for leaping box forecasts by using voting procedures, which further raises the reliability of the ensemble.

TPH-YOLOv5 is proposed by Zhu et al. [19]. To better recognise objects of varying sizes, we augment YOLOv5 with an additional prediction head. Next, we swapped out the standard prediction heads with to test out the device's impact on prediction accuracy. To locate the attention region in settings with many items, we additionally use the convolutional block attention model (CBAM). Our suggested TPH-YOLOv5 may be further enhanced by using the many techniques we present, including as data augmentation, testing, multi-model integration, and the introduction of an additional classifier. Extensive trials using the VisDrone2021 dataset demonstrate that TPH-YOLOv5 achieves high performance with remarkable interpretability in drone-captured circumstances. TPH-AP YOLOv5's results on DET-test-challenge are 39.18%, which is an improvement of 1.81% over the previous SOTA technique (DPNetV3). After competing in the VisDrone Challenge 2021, TPHYOLOv5 came in fifth place, with results that were quite similar to those of the winning model (AP 39.43%). TPH-YOLOv5 outperforms the basic model (YOLOv5) by roughly 7%, which is promising and in line with current market standards.

Ways to improve object detection performance in such scenarios are investigated by Jung and Choi [20]. The conditions under which the photographs were taken made it difficult to identify any particular object. The experimental data was collected through the use of images taken under a variety of scenarios, including those where the drone's height was altered, where there was no available light, and so on. The F11 4K PRO drone and the VisDrone dataset were used to capture all of the experimental data. As a result of this research, we offer forth some suggestions on how the YOLOv5 model may be made more efficient. In order to determine the most important metrics, we fed them into both the standard YOLOv5 model and our own, revised YOLOv5 Ours. When compared to the original YOLOv5 model, which are the primary indications. In the end, we drew our conclusion from the information we gathered by contrasting the baseline YOLOv5 model with our own, refined YOLOv5 model. Our investigation led us to a conclusion on the optimal model for object detection in a wide range of scenarios.

Multi-Proxy Recognition Network with Packing (UFPMP-Det) is a new technique for object detection on drone footage proposed by Huang et al. [21]. To handle the plethora of extremely tiny scales, the Unified Foreground Packing (UFP) module first merges the sub-regions provided by a coarse detector by clustering to suppress contextual, and then packs the resultant ones into a mosaic for a single infrared detector. Extensive experiments on the VisDrone and UAVDT datasets demonstrate that UFPMP-Det can rapidly generate new state-of-the-art scores.

After capturing aerial pictures using a rotorcraft drone based object recognition to identify trees that may be infected with pine wilt disease [22]. In each of the obtained multispectral aerial pictures, you'll see spectral bands for the visible spectrum, near infrared light, the green spectrum, the red spectrum, and the red edge. Aerial images were subjected to image. After that, the multichannel CNN-based object discovery was trained and tested using a massive quantity of data gathered through data augmentation. Excellent discovery consequences were achieved with mAP 86.63% and regular connection over union 71.47% after validating the detection presentation of the trained perfect.

Using the AU-AIR dataset, Gupta and Verma [23] describe innovative techniques for monitoring and surveillance of aerial images of traffic, based on widely-used DL object identification models. This dataset is quite unbalanced, thus 500 more pictures were harvested using web-mining methods to even things out. This study makes a unique contribution in two ways. To begin, this article provides a rigorous scientific explanation for why photographs taken from the ground can't be used for detecting objects in the sky. To further examine the efficacy of these algorithms, a regress comparison was

performed. Extensive experimental investigation verifies YOLOv4's efficacy, showing that it surpasses its competitors by at least 88 percent in terms of mean absolute performance (mAP). As an added bonus, its real-time practical application is ensured by its detection speed, which is more than six times as fast as before, as well as its flexibility and detection resilience.

2.1 Challenges

Designing a deep neural network, developing a real-time tracking method, and implementing a safeguard against interference are all difficult tasks. When developing a deep neural network, it is crucial to carefully consider both the sum of layers and the dataset used. Identifying and following a person takes significantly less time using our technique.

Designing a system that works without being disrupted by outside factors is another obstacle. It can take a long time to settle on a suitable threshold value. Those other difficulties include:

- solution of occlusion in Camshift algorithm
- solution of lighting in the picture improvement of system speed
- development of deep neural network construction

3. PROPOSED SYSTEM

3.1 Datasets

For this research, we used two different sets of UAV images, each of which depicts a different environment with a wide variety of objects ranging in size and shape. UAV multiple objects present are scarce compared to satellite image datasets. The quantity of data, variety of objects, camera angles, lighting scenarios, and geographic locations all played a role in our decision to use these UAV datasets.

3.1.1 VisDrone dataset

For drone-based requests and autonomous navigation, researchers in computer vision have been interested in improving methods for object recognition in UAV photos. The VisDrone data sets were developed to aid investigation into this area. This dataset posed a problem for OD and tracking researchers, so they used ensemble detection methods and state-of-the-art algorithms to solve it. DPNet-ensemble, RRNet, and ACM-OD were the best three detectors, all attaining 29.13% Aps or above. Real-world applications highlight the need for advancement in this field, as the top detector DPNet-ensemble scored less than a 30% AP score. The VisDroneDET2019 Dataset is identical to the VisDrone-DET2018 Dataset in that both were acquired using drone platforms to collect a total of 8599 photos from a variety of locations and elevations. In all, there are 540k leaping boxes of target objects annotated, spread over 10 categories. Transportation modes include: awning-tricycle, bicycle, pedestrian, vehicle, and tricycle. To facilitate training and evaluation, the dataset: 6471 photos for training, 548 for validation, and 1610 for testing, all of which were captured in one of 14 locations across China but feature distinctly diverse settings. All photos used are 1360 pixels wide by 765 pixels tall at input. Images in the collection have a maximum resolution of 2000 by 1500 pixels. The detectors' efficacy has been evaluated using the test dev set of 1610 photos.

3.1.2 AU-AIR dataset

This multi-modal UAV dataset (Aarhus, Denmark) has UAV photos from 2 hours (8 video streams) of traffic shadowing on Skejby Nordlandsvej and P.O. Pedersensvej. UAVs were employed to record footage for the dataset, and the films show a range of flying altitudes, from 10 metres to 30 metres, and camera angles, from 45 degrees to 90 degrees. The photos used as input have a resolution of 1920 pixels by 1080 pixels. Images in the collection are up to 1920 by 1080 in size. Images captured in sunny, overcast, and partly cloudy situations are all represented in the dataset. People, automobiles, buses, vans, trucks, bicycles, motorcycles, and trailers are only some of the eight object classes represented in the dataset, with just three of these classes heavily represented in the annotated bounding boxes. In comparison to the benchmark networks, which each reached 30.22 mAP, the latter only managed 19.50 mAP. A total of 4000 photos were generated by augmenting 1000 images from this dataset, and those were used to evaluate the findings.

3.1.3 Handling the dataset challenges

Due to factors such as large images and a lack of available drone datasets, object detection in UAV images is challenging. Data augmentation and the suggested methods are a good way to deal with these problems.

3.2 Data augmentation

Researchers offered novel data augmentation strategies for optimal biodiversity discovery in the wild, such as making several rotational copies of the original picture, flipping it horizontally and vertically, mirroring it, rotating it, and moving it horizontally and vertically. Researchers commonly utilise techniques such as histogram equalisation, Gaussian blur, random translation, scaling, cut off, and rotation on UAV datasets for usage in other applications like vehicle and OD. Table 1 details all of the data enhancement methods that were employed in this analysis.

Table 1. Data rise practices used in the study

Average blurring	None
Balance histogram procedure	Rotation by 10◦
Flipping the image steeply	Rotation by 90◦
Changing to HSV color space	Levitation the hue
Blurring the copy	Raising the red station
Cropping the copy	Raising the saturation
Dropout	Levitation the rate
Elastic misrepresentation	Resizing the image
Bilateral blurring	Levitation the blue station
Blurrin	Raising the green channel
Tossing the image horizontally	Rotation by 180◦
Tossing the image horizontally	Rotation by 270◦
Applying Gamma alteration	Addition salt and pepper noise
Gaussian blurring	Sharpen the image
Upsetting the image	Shearing image

3.3 The target tracking based on Camshift

The Camshift procedure, which is an improvement on the Meanshift algorithm, is able to monitor the distribution of probabilities for changes in the environment. The algorithm's central idea is to apply the Meanshift operation to each video image in the sequence, and then use the information gleaned from the preceding frame's Meanshift operation to inform the current frame's. The colour probability distribution derived

from the histogram back projection is specifically used as a reference point. By modifying the window's size and position, it's possible to zero in on the action in the current video frame.

1) Back projection. The H component of a picture is counted after converting video frames from RGB to HSV colour space in order to produce a consistent quantized colour histogram of the H component. Next, the histogram's corresponding colour probability lookup table is computed, and the probability of the occurrence of the colour of the point is substituted for the colour map.

2) Meanshift iteration. Success in reaching the target location is achieved through gradient optimization of the probability delivery, which yields the distribution's peak value. The following is a diagrammatic representation of the algorithm's development process:

$$M_{00} = \sum_x \sum_y I(x, y) \tag{1}$$

$$M_{01} = \sum_x \sum_y yI(x, y) \tag{2}$$

$$M_{10} = \sum_x \sum_y xI(x, y) \tag{3}$$

In order to determine where the centre of mass is located, we need to first determine its mass components (M_{00}, M_{01}, and M_{10}) using the above formula.:

$$(x_c, y_c) = \left[\frac{M_{10}}{M_{00}} \frac{M_{01}}{M_{00}}\right] \tag{4}$$

On this foundation, the scope of the search window is attuned rendering to formula (5):

$$S = 2 \times \sqrt{M_{00}/256} \tag{5}$$

The distance is measured from the preliminary point to the target location to see if it is greater than the threshold you've selected for the search window's centre. In such case, you'll need to recalculate the window's updated centre of mass and make the necessary adjustments to its location and size. In order to reposition the target, the next frame's image is read once the convergence condition is met, which occurs when the sum of iterations spreads the maximum allowed or the distance travelled is less than a threshold value.

3) Camshift. In continuous video frames, the Camshift procedure is an extension of Meanshift. Each video frame undergoes a Meanshift operation, with the resultant data from that frame being used to seed the current frame's adaptively adjusted search window's size and position. Eventually, this will allow for the desired tracking and location.

3.4 Object detection network module in the projected deep perfect

There are two parts to the object identification network module in the projected deep model; the first is responsible for gathering features, while the second is responsible for combining them into a single set. The following is an explanation of the two modules.

First, an object detection network module that takes the color-converted picture as input and extracts relevant features therefrom. Feature extraction CNN units (labelled "Feature extraction Conv" in Figure 1), receptive layers are cascaded to form the network module responsible for feature extraction, which is described in further detail below. Figure 2 depicts the layers that make up. Extracting primary features from an input feature map that is just half as large spatially is the primary function of the feature extraction CNN module. Our feature extraction CNN module only uses a small sum of feature maps, which drastically reduces the computational overhead of the convolutional processes. And as can be seen in Figure 3, uses a 1x1 convolution operation with a 1x1x32 kernel size to effectively double the channel size. At this early stage, increasing the feature maps is the primary focus in preparation for the feature extraction phase. Conversely, the 1x1 convolution process is utilised to half the channel size in each used in the subsequent stages, hence reducing the feature sizes and the computational difficulty. For instance, the 1x1 convolution layer used in the second stage uses a kernel size of 1x1x64 to divide the feature channels by two. The responsive module is proposed to further refine the features from the preceding, mitigating the risk of inadequately extracted features. Using the responsive module, we can extract multiscale features from the input feature map quickly and easily. The primary objective is to maintain the feature representational power while minimising computing complexity, as shown below. Layers of expansion and compression as well as the shorter connection are all part of the proposed receptive module, as exposed in Figure 2. The inception v3 module depends on appropriately factorised convolutions to make the most of the increased processing needed to develop the network, allowing for spatial aggregation across without a significant loss in representational capacity. Two convolution operations of size 33 were carried out by a convolution operation of size 55 in the inception module before this one [24-26]. It was mostly from the inception v3 that we stole the concept of factorization of convolution operations to apply in our expand layer, since this has been demonstrated to be successful for decreasing computational complexity.

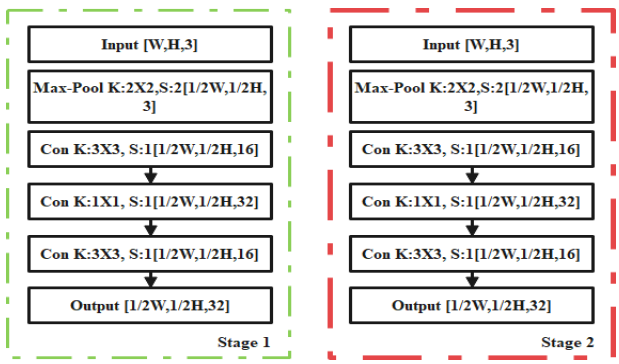

Figure 1. Projected model for object finding

For each layer, [W, H, C] refers to the dimensions of the resulting feature map in terms sum of channels (C). Further, Conv stands for the convolution process and Max-pool for the maximum pooling operation. Also, x represents the input feature map, and R(x) is the result of the Crush layer's operation on x.

The architecture of the extend layer is different from that of Inception v3, and it is also easier to understand. Extending the processing can be used to recognise larger or smaller substances in an image. After the expand layer, a squeeze layer uses a single 1x1 convolution operation to reduce the number of feature channels. The input feature map of a receptive module may be maintained, and the vanishing gradient problem can be avoided by using a shortcut link to transmit the

input, which is then concatenated with the output of the squeezing layer to generate the receptive module's output, which has been demonstrated to outperform ResNet in feature preservation, which inspired the use of the chain operation to generate the output, as opposed to the element-wise addition employed in ResNet. The output of the receptive module is then convoluted with a 3x3 kernel and a stride size of 1.

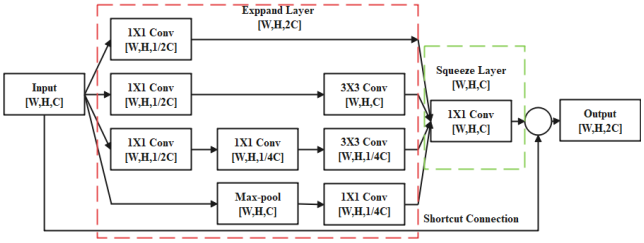

Figure 2. Receptive unit in the projected feature extraction system component

There are five nested levels of processing in the feature extraction network component. Each of the initial four steps includes a feature extraction convolution layer. Only one CNN is used for feature extraction in the final step. The feature aggregation network module, which will receive the results of the last three steps, is explained below. FPN's primary idea is to construct feature pyramids at low additional cost by capitalising on deep CNN's inbuilt multiscale pyramidal hierarchy. Extraction of high-level semantic feature maps at all sizes relies on the development of a top-down construction with lateral linkages. Lateral connection blocks are made to combine feature maps with the same spatial resolution from both the bottom-up path and the top-down path.

An make up the lateral connection block. Our version of the proposed network module sends feature maps from stages 3, 4, and 5 to the feature aggregation network module. In the end, the feature aggregation network will generate three distinct outputs for object detections at varying sizes; this means that, using the learnt may be produced independently at any level. In addition, the hyper-parameters such as momentum, learning rates and epochs are optimally selected using GOA.

3.4.1 Gannet Optimization Algorithm (GOA)

We present a novel meta-heuristic optimization method we name the gannet optimization algorithm, which takes its inspiration from the gannet's predatory nature. To model the predatory actions of pond geese, we present an optimization system with two phases: exploration and exploitation. mode, abrupt rotation, and random wandering are the four distinct forms of predatory behaviour that may be seen throughout the exploration and exploitation phases.

Initialization phase

As shown in Eq. (6), the GOA begins with a collection of random solutions, from which the best one is selected as the best global solution.

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x1_{,Dim-1} & x_{1,Dim} \\ x_{2,1} & \cdots & x_{2,j} & \cdots & x2_{,Dim-1} & x_{2,Dim} \\ \cdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & x_{i,j} & \cdots & \cdots & \cdots \\ x_{N-1,1} & \cdots & x_{N-1,j} & \cdots & \vdots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N-1,Dim-1} & x_{N-1,Dim} \\ & & & & x_{N,Dim-1} & x_{N,Dim} \end{bmatrix} \quad (6)$$

x_i signifies the position of the ith separate. Each $x_{i,j}$ in the matrix X can be intended by Eq. (7).

$$x_{i,j} = r_1 \times \left(UB_j - LB_j\right) + LB, i = 1, 2, \ldots, N, j = 1, 2, \ldots, Dim \quad (7)$$

where, N is the total sum of people in the populace, Dim is the scope of the problematic in dimensions, and r_1 is a chance sum among 0 and 1.

Additionally, the memory matrix, an MX matrix, is defined. During setup, the X matrix's values are transferred to MX. The memory matrix MX will keep track of the gannet individuals' shifting positions as the evolutionary process repeats. If the fitness function determines that an individual in the memory solution.

Exploration phase

From above, gannets look for their food in the water, and once they locate it, they dive at an angle that corresponds to the depth to which their catch has sunk. For the U-shaped dive, we use Eq. (9), and for the V-shaped dive, we use Eq. (10).

$$t = 1 - \frac{It}{T \max\_iter} \quad (8)$$

$$a = 2 * \cos(2 * \pi * r_2) * t \quad (9)$$

$$b = 2 * V(2 * \pi * r_3) * t \quad (10)$$

$$V(x) = \begin{cases} -\frac{1}{\pi} * x + 1, x \in (0, x) \\ \frac{1}{\pi} * x - 1, x \in (\pi, 2\pi) \end{cases} \quad (11)$$

where, $T\ max\_iter$ is the supreme sum of iterations, r_2 and r_3 are random statistics among 0 and 1, and it is the current number of iterations.

Utilizing position updating is the next step. We define a random amount q to select between the two dive strategies, since the probability of a gannet selecting one over the other when they are predating is very close to 1. Position inform formula is shown in Eq. (12).

$$MX_i(t+1) = \begin{cases} X_i(t) + u1 + u2, & q \geq 0.5 \\ X_i(t) + v1 + v2 & q < 0.5 \end{cases} \quad (12)$$

$$u2 = A * (X_i(t) - X_r(t)) \quad (13)$$

$$v2 = B * (X_i(t) - X_m(t)) \quad (14)$$

$$A = (2 * r_4 - 1) * a \quad (15)$$

$$B = (2 * r_5 - 1) * b \quad (16)$$

where, $X_i(t)$ is the ith member of the current population, $X_r(t)$ is a randomly, $X_m(t)$ represents the average position of members of the current population, and $X_m(t)$.

$$X_m(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t) \quad (17)$$

Exploitation phase

Once the gannet has rushed into the water in either of the above two ways, there are two further steps that must be taken to maximise exploitation. Skilful fish in the water typically make a sharp turn to evade a gannet's pursuit. A lot of effort is put in by the gannet in order to catch the fish that are desperately attempting to get away. Capture capacity is defined in this context using Eq. (18). The gannet successfully catches a fish when it has plenty of energy and a large capture capacity, click here. When the gannet's energy gradually declines and it is unable to finish the capturing motion.

$$Capturability = \frac{1}{R*t2} \quad (18)$$

$$t2 = 1 + \frac{It}{Tmax\_iter} \quad (19)$$

$$R = \frac{M*vel^2}{L} \quad (20)$$

$$L = 0.2 + (2 - 0.2) * r_6 \quad (21)$$

where, r_6 is a random sum among zero and one, M = 2.5 Kg is the weight of the gannet, and Vel = 1.5 m/s is the gannet's speed in the water, disregarding the resistance of the water for the time being.

If the gannet's grasp is within striking distance of its prey, the position is updated with a sharp turn; if not, the gannet performs a Levy, see Eq. (22)

$$MX_i(t+1) = \begin{cases} t * delta * \left(X_i(t) - X_{Best}(t)\right) + X_i(t), & Capturability \geq c \\ X_{Best}(t) - \left(X_i(t) - X_{Best}(t)\right) * P * t, & Capturability < c \end{cases} \quad (22)$$

$$delta = Capturability * |X_i(t) - X_{Best}(t)| \quad (23)$$

$$P = Levy(Dim) \quad (24)$$

where, c=0.2 is a constant whose value was settled on after extensive experimentation, $X_{Best}(t)$ is the top performer in the current population, of the Levy distribution.

$$Levy(Dim) = 0.01 \times \frac{\mu \times \sigma}{|v|^{\frac{1}{\beta}}} \quad (25)$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\frac{(1+\beta)}{2} \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \quad (26)$$

4. RESULTS AND DISCUSSION

Python 3.7, PyTorch 1.1.0, NumPy 1.16.2, NetworkX 2.4, and 2.1.0 were used to create the proposed model. The tests were conducted on a computer equipped with an 8 GB GeForce RTX 2070, a 7th generation 32 GB of RAM.

4.1 Performance metrics

Every sample is assigned a predicted label based on the classification model's predictions. As a result, each sample is classified into one of the following four groups:

- Authentic positives that are properly foretold positives are named true positives (TP);
- Authentic positives that are erroneously forecast negatives are named false negatives (FN);
- Authentic negatives that are properly forecast negatives are named true negatives (TN);
- Authentic negatives that are imperfectly forecast positives are named false positives (FP).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (27)$$

$$F_1 score = \frac{2.TP}{2.TP+FP+FN} \quad (28)$$

$$Precision = \frac{TP}{TP+FP} \quad (29)$$

$$Recall\ /\ Sensitivity = \frac{TP}{TP+FN} \quad (30)$$

Table 2 presents the experimental analysis of projected model with existing techniques on VisDrone dataset. The mentioned existing techniques are considered and implemented on our system, then results are averaged in the below table.

Table 2. VisDrone results on various techniques

Network type	ACC	SEN	SPEC	F-MEASURE	FPR
ETL [18]	0.84	0.55	0.99	0.70	0.56
TPH-YOLOv5 [19]	0.88	0.62	0.99	0.76	0.53
YOLOv5 [20]	0.87	0.65	0.99	0.78	0.50
UFPMP-Det [21]	0.93	0.75	0.99	0.85	0.49
Proposed	0.95	0.79	0.99	0.89	0.48

In the above table represent that the VisDrone 2019 Test-dev set Results. we have compared the proposed model with different model as ETL [18], TPH-YOLOv5 [19], YOLOv5 [20] and UFPMP-Det [21]. But this comparisons analysis, the proposed model reaches the better accuracy of 0.95 respectively. Figure 3 and 4 represent the graphical analysis.

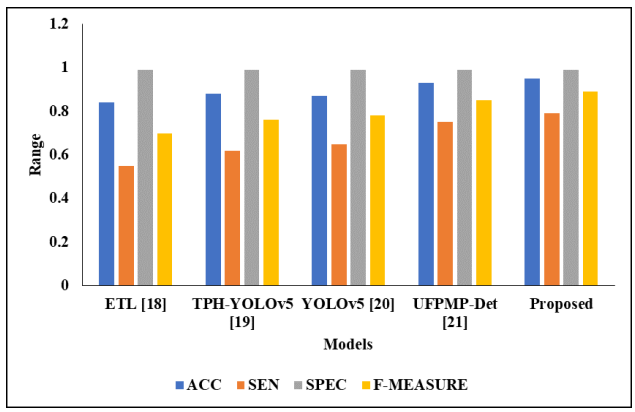

Figure 3. Comparative analysis of projected perfect with existing techniques

Figure 3 represents that the Proportional Analysis of projected model with existing techniques. In this analysis, the proposed model reached the better results than another comparing model.

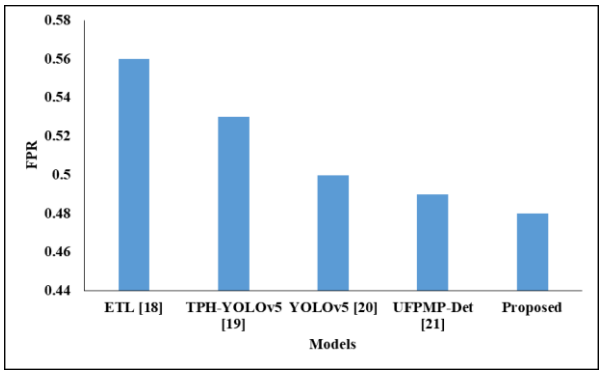

Figure 4. FPR comparison

Figure 4 represents that the FPR Comparative Examination of projected model with existing techniques. In this analysis, the proposed model reached the better FPR comparison results than another comparing model.

Table 3. AU-AIR dataset results on various techniques

Network Type	ACC	SEN	SPEC	F-MEASURE	FPR
ETL [18]	0.78	0.46	0.99	0.63	0.62
CNN [22]	0.85	0.56	0.99	0.70	0.58
Faster R-CNN [23]	0.82	0.57	0.99	0.78	0.52
YOLOv4 [23]	0.95	0.89	0.99	0.90	0.46
Proposed	0.97	0.89	0.99	0.90	0.46

Table 3 represents that the AU-AIR Dataset Results. We have compared the projected model with different model as ETL [18], CNN [22], Faster R-CNN [23] and YOLOv4 [23]. But this comparisons analysis, the proposed model reaches the better accuracy of 0.97, respectively. Figures 5 and 6 represent the graphical comparison between various techniques.

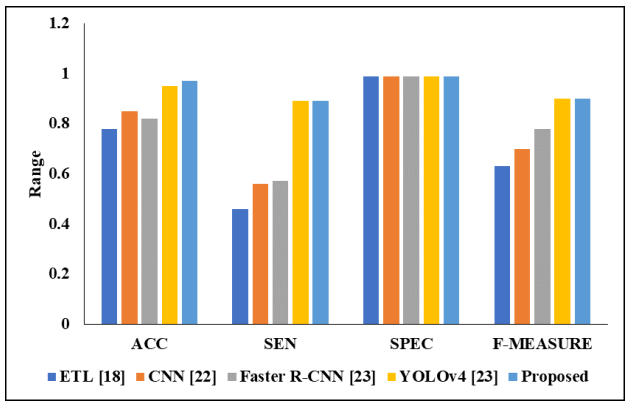

Figure 5. Graphical representation of projected perfect with existing procedures

Figure 5 represents that the Graphical Representation of projected model with existing techniques. In this analysis, the proposed model reached the better comparison results than another comparing model.

Figure 6 represents that the Graphical Representation of proposed model with existing techniques FPR comparison. In this analysis, the proposed model reached the better FPR comparison results than another comparing model.

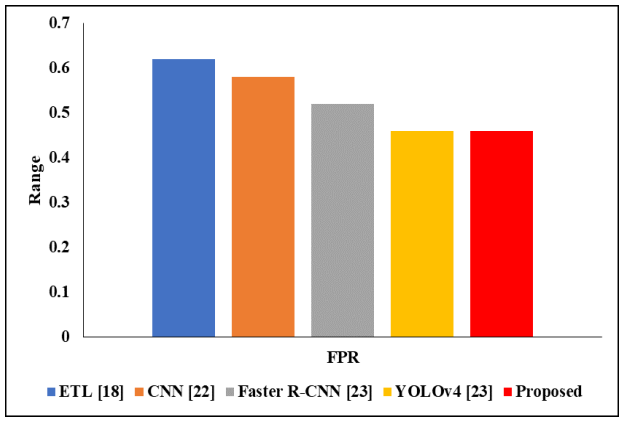

Figure 6. FPR comparison

5. CONCLUSIONS

We suggest a small, lightweight, end-to-end deep neural network for detecting objects based on GOA. We performed comprehensive testing and analysis of drone-based picture datasets' proposed model with augmentation strategies. Based on how well they worked on the selected datasets, the optimised model and augmentation methods show promise for UAV object recognition. Several test train augmentation methods have shown promise in alleviating the shortage of UAV picture datasets. The proposed model detector with colour augmentation achieves a performance of 95% AP for detecting bicycles on the VisDrone dataset and a performance of 97% AP for detecting buses on the AU-AIR dataset.

5.1. Future scope

We've seen that this strategy falls short when it comes to spotting novel things like the awning tricycle and the tricycle, both of which weren't included in the training datasets. More models will be incorporated into the assembly process, and the algorithm will be tested on more drone-based datasets in future studies. The multiscale object identification approach suggested in this study for drone-based object finding may also be used to produce higher-quality orthomosaics, particularly for items located in the image's periphery.

REFERENCES

[1] Hariharan, B., Arbeláez, P., Girshick, R., Malik, J. (2014). Simultaneous detection and segmentation. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, pp. 297-312. https://doi.org/10.1007/978-3-319-10584-0_20

[2] Hariharan, B., Arbeláez, P., Girshick, R., Malik, J. (2015). Hypercolumns for object segmentation and fine-grained localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 447-456.

[3] Othman, N.A., Aydin, I. (2021). Challenges and Limitations in Human Action Recognition on Unmanned Aerial Vehicles: A Comprehensive Survey. Traitement du Signal, 38(5): 1403-1411. https://doi.org/10.18280/ts.380515

[4] Vatambeti, R., Mantena, S.V., Kiran, K.V.D., Manohar, M., Manjunath, C. (2023). Twitter sentiment analysis on

online food services based on elephant herd optimization with hybrid deep learning technique. Cluster Computing, 1-17. https://doi.org/10.1007/s10586-023-03970-7

[5] Karpathy, A., Li, F.F. (2015). Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3128-3137.

[6] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In International Conference on Machine Learning, pp. 2048-2057.

[7] Kailasam, S., Achanta, S.D.M., Rama Koteswara Rao, P., Vatambeti, R., Kayam, S. (2022). An IoT-based agriculture maintenance using pervasive computing with machine learning technique. International Journal of Intelligent Computing and Cybernetics, 15(2): 184-197. https://doi.org/10.1108/IJICC-06-2021-0101

[8] Kang, K., Li, H., Yan, J., Zeng, X., Yang, B., Xiao, T., Ouyang, W. (2017). T-CNN: Tubelets with convolutional neural networks for object detection from videos. IEEE Transactions on Circuits and Systems for Video Technology, 28(10): 2896-2907. https://doi.org/10.1109/TCSVT.2017.2736553

[9] Tripathi, S., Kang, B., Dane, G., Nguyen, T. (2017). Low-complexity object detection with deep convolutional neural network for embedded systems. In Applications of Digital Image Processing XL, 10396: 317-331. https://doi.org/10.1117/12.2275512

[10] Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, pp. 2980-2988.

[11] De Bruin, A., Booysen, T. (2015). Drone-based traffic flow estimation and tracking using computer vision: transportation engineering. Civil Engineering=Siviele Ingenieurswese, 2015(8): 48-50.

[12] Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D. (2009). Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9): 1627-1645. https://doi.org/10.1109/TPAMI.2009.167

[13] Alharbi, L.M., Qamar, A.M. (2022). Arabic Sentiment Analysis of Eateries' reviews using deep learning. Ingénierie des Systèmes d'Information, 27(3): 503-508. https://doi.org/10.18280/isi.270318

[14] Zhu, P., Wen, L., Bian, X., Ling, H., Hu, Q. (2018). Vision meets drones: A challenge. arXiv preprint arXiv:1804.07437. https://doi.org/10.48550/arXiv.1804.07437

[15] Zhu, P.F, Wen, L.Y., Du, D.W., Bian, X., Hu, Q.H., Ling, H.B. (2020). Vision meets drones: Past, present and future. arXiv 2020, arXiv:2001.06303. https://doi.org/10.48550/arXiv.2001.06303

[16] Bozcan, I., Kayacan, E. (2020). Au-air: A multi-modal unmanned aerial vehicle dataset for low altitude traffic surveillance. In 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, pp. 8504-8510. https://doi.org/10.1109/ICRA40945.2020.9196845

[17] Roy, A.M., Bose, R., Bhaduri, J. (2022). A fast accurate fine-grain object detection model based on YOLOv4 deep neural network. Neural Computing and Applications, 34: 1-27. https://doi.org/10.1007/s00521-021-06651-x

[18] Walambe, R., Marathe, A., Kotecha, K. (2021). Multiscale object detection from drone imagery using ensemble transfer learning. Drones, 5(3): 66. https://doi.org/10.3390/drones5030066

[19] Zhu, X., Lyu, S., Wang, X., Zhao, Q. (2021). TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2778-2788.

[20] Jung, H.K., Choi, G.S. (2022). Improved yolov5: Efficient object detection using drone images under various conditions. Applied Sciences, 12(14): 7255. https://doi.org/10.3390/app12147255

[21] Huang, Y., Chen, J., Huang, D. (2022). UFPMP-Det: Toward accurate and efficient object detection on drone imagery. In Proceedings of the AAAI Conference on Artificial Intelligence, 36(1): 1026-1033. https://doi.org/10.1609/aaai.v36i1.19986

[22] Park, H.G., Yun, J.P., Kim, M.Y., Jeong, S.H. (2021). Multichannel object detection for detecting suspected trees with pine wilt disease using multispectral drone imagery. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14: 8350-8358. https://doi.org/10.1109/JSTARS.2021.3102218

[23] Gupta, H., Verma, O.P. (2022). Monitoring and surveillance of urban road traffic using low altitude drone images: a deep learning approach. Multimedia Tools and Applications, 81(14): 19683-19703. https://doi.org/10.1007/s11042-021-11146-x

[24] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS. IEEE transactions on pattern analysis and machine intelligence, 40(4): 834-848. https://doi.org/10.1109/TPAMI.2017.2699184

[25] Deore, S.P. (2022). Human behavior identification based on graphology using artificial neural network. Acadlore Transactions on AI and Machine Learning, 1(2): 101-108. https://doi.org/10.56578/ataiml010204

[26] Shao, X.H., Chang, D.F., Li, M.J. (2022). Optimization of lateral transfer inventory of auto spare parts based on neural network forecasting. Journal of Intelligent Systems and Control, 1(1): 2-17. https://doi.org/10.56578/jisc010102