ICASSP 2023

4 - 10 JUNE,
RHODES ISLAND, GREECE

SETNET: A SPARSE ENSEMBLE NETWORK FOR DRONE LOCALIZATION AND ZERO SHOT DRONE TRACKING IN REAL TIME SURVEILLANCE VIDEOS

Dharini Raghavan, S Sethu Selvi

Department of Electronics and Communication Engineering, Ramaiah Institute of Technology, India

4th June 2023

RAMAIAH
Institute of Technology

1. OVERVIEW

Unmanned Aerial Vehicles (UAVs)

- ***Applications*** – search and rescue (SAR) operations, disaster management, remote sensing, traffic monitoring, war reporting, surveillance in military and airline operations
- ***Proliferation*** – serious security threat, privacy concerns
- ***Research gaps*** **–** efficient target localization and tracking in multitude of environmental and topographical conditions, dynamic backgrounds

Challenges in Drone Localization and Tracking

- Unfavorable topographical conditions – long-range target detection, uneven illumination, weak background contrast, environmental distortions, close resemblances to birds – higher probability of ***false alarms***

Existing Methods for Target Localization and their Limitations

- ***Multimodal approaches*** **–** radar, radio frequency (RF), acoustic sensing and Lidar
- ***Limitations*** – expensive, energy inefficient, not being deployable in noisy environments, sophisticated infrastructure for integration with UAVs
- Fail to differentiate between drones and birds at long ranges
- Do not achieve robust drone localization under extreme topographies, low visibility conditions and distorted environmental scenarios
- Computer vision and video analytics – ***promising modality***, low visibility and unfavorable conditions, dynamic backgrounds

2. RELATED WORK

Saqib, M et. al, "A Study on Detecting Drones Using Deep Convolutional Neural Networks," In Proceedings of the 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August – 1 September 2017, pp. 1 – 5.

Transfer learning approach, combination of CNN and VGG-16, VGG-16 along with Faster R-CNN outclassed other networks – MPEG4 coded videos with drones

Shi, Q et. al, "Objects Detection of UAV for Anti-UAV Based on YOLOv4," In Proceedings of the 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Weihai, China, 14 – 16 October 2020, pp. 1048 – 1052.

YOLOv3 and YOLOv4 compared for UAV detection at low altitudes, YOLOv4 reported to have higher accuracy and inference speed – custom dataset consisting of three different categories of drones namely: DJI-Phantom, DJI-Inspire, XIRO-Xplorer

Liu, H et. al, "Real-Time Small Drones Detection based on Pruned YOLOv4," Sensors 21, no.10: 3374, 2021.

Pruning of YOLOv4 architecture – thinner and shallower, pruned version of YOLOv4 with a channel prune rate of 0.8 and 24 pruning layers – mAP score of 90.5%, improvement of 60.4% in processing speed

B. Taha et. al, "Machine Learning-Based Drone Detection and Classification: State-of-the-Art in Research," in IEEE Access, vol. 7, pp. 138669- 138682, 2019

Drone detection and classification using machine learning algorithms with different modalities like radar, visual, acoustic, and radio-frequency sensing systems – accuracy of machine learning algorithms trained on visual technologies (images/videos) – significantly better than other modalities

3. CONTRIBUTIONS

Sparse Ensemble Tracker Network (SETNET)

1. Distinct Features

- Ensemble of base YOLOv5 networks (YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x)
- ***Model compression*** – static pruning and quantization
- ***Model optimization*** – hyper-parameter evolution-based genetic algorithm – to improve model generalization
- ***Model ensembling*** – non-maximum suppression algorithm
- ***Tracker network*** – Contrastive Language Image Pre-training (CLIP)-based zero shot drone tracking algorithm – assigns a unique ID to drones spotted in video instances, helps track them using feature similarity

2. Benchmark Evaluation

- An overall improvement in small ***target localization*** and robust ***trajectory tracking***
- ***Five-fold*** improvement in inference speed – suitable for real time deployment in resource constrained environments
- Evaluated under a range of background distortions and scenarios
- Compared with several ***state-of-the-art algorithms*** – outperforms both in terms of accuracy of localization and inference speed

4. METHODS AND EXPERIMENTS

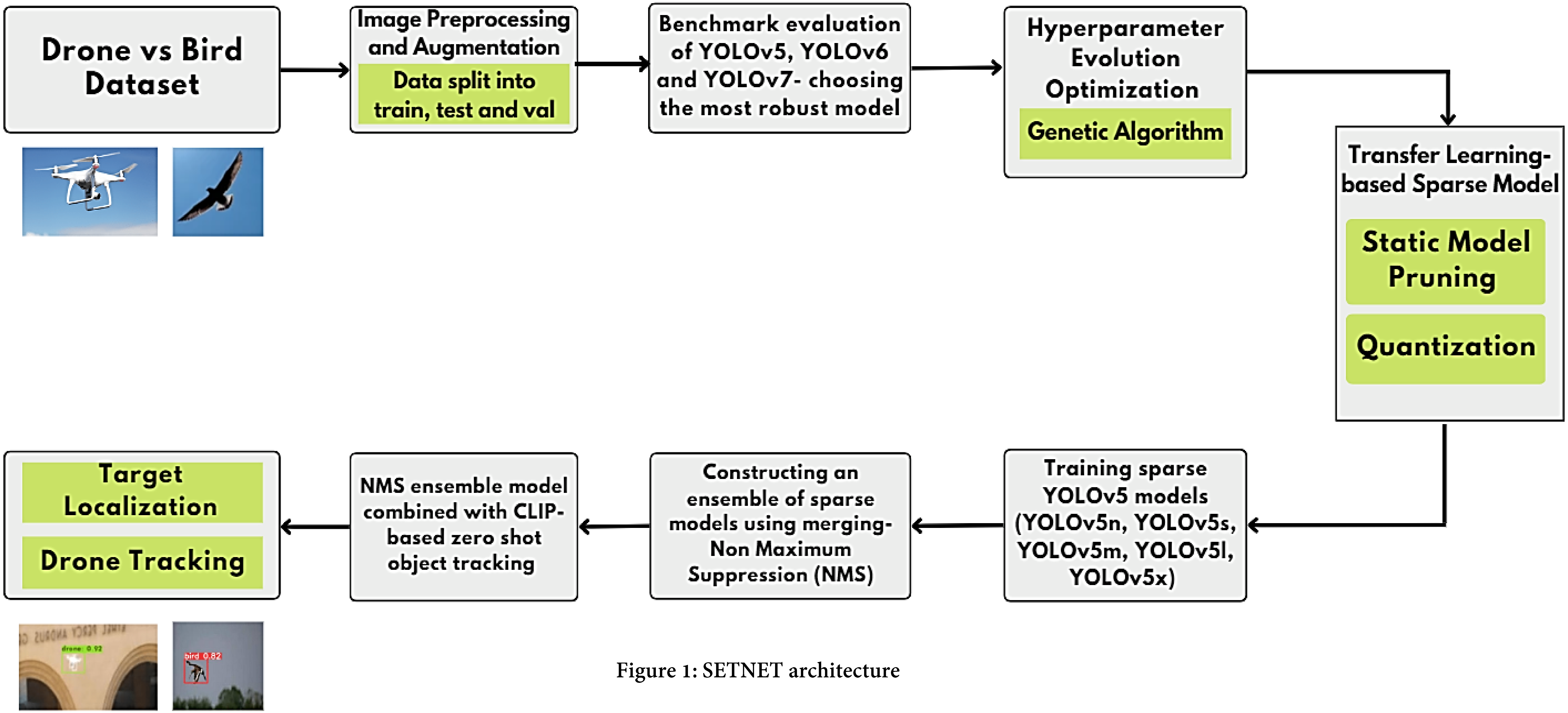


Figure 1: SETNET architecture

4. METHODS AND EXPERIMENTS

Drone and Bird Dataset

Extensive dataset – Birds, different categories of drones such as quadcopters, hexa-rotors, octa-rotors curated from several sources

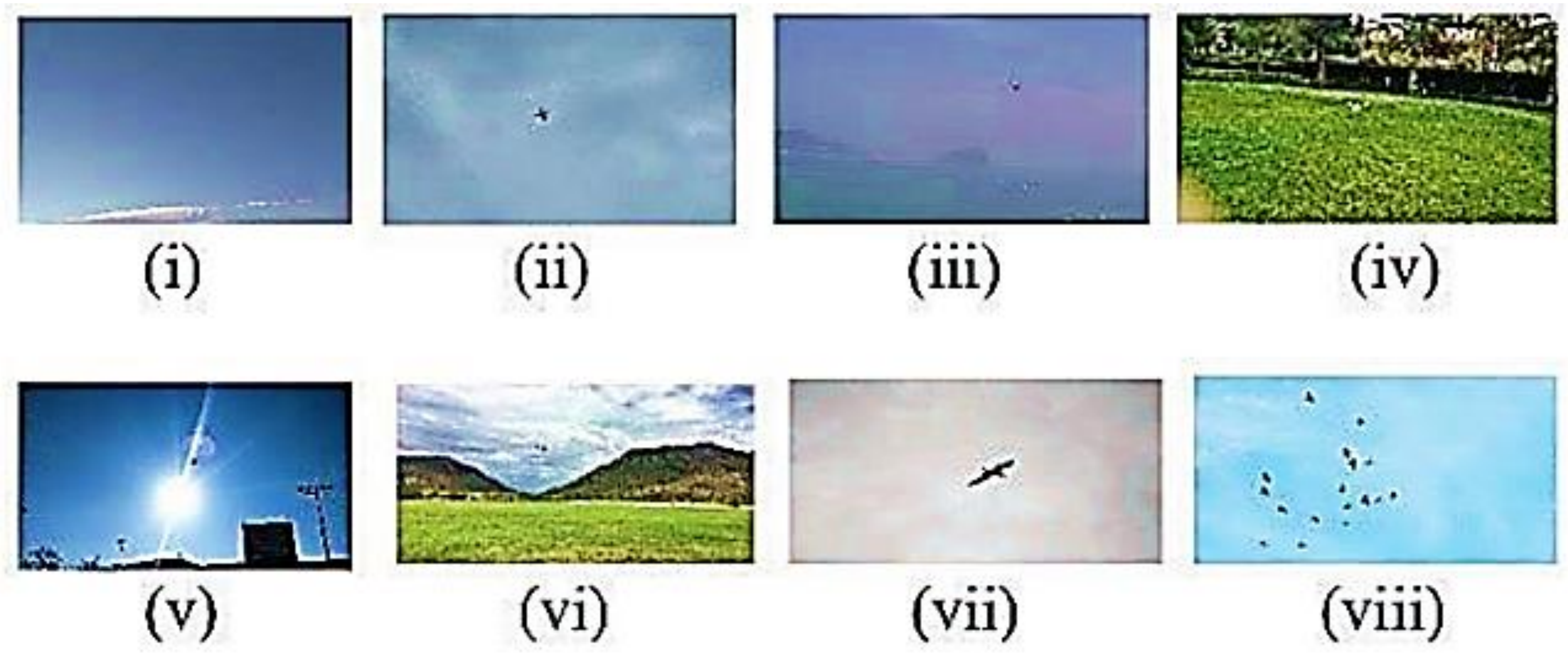


Figure 2: (i) targets at long range (ii) targets camouflaged by clouds (iii) low visibility due to mist (iv) targets camouflaged by background tree cover (v) uneven illumination (vi) unfavorable topography (vii) bird resembling a drone (viii) swarm of drones and birds

Table 1: Dataset description

Parameter Considered	Description
Number of classes	2 (Drone and Bird)
Data split ratio	70:10:20 (train:validation:test)
Preprocessing	Auto orient, static crop and image resize
Augmentation	Flips (horizontal and vertical), mosaic, neural style transfer, rotation, gamma correction, contrast stretching, histogram equalization
Image size	640×640
Environmental Factors	Hilly regions, thick forest cover, uneven illumination, cloudy sky, fog and mist
Scenarios	Single class in a frame, multiple classes in a frame, objects far-off from the FoV of the source camera, swarm of drones and birds
Distortions	Salt and Pepper noise, Gaussian blur, camera distortions, AWGN

4. METHODS AND EXPERIMENTS

Data Augmentation

Data augmentation: image flip, rotation by various angles, gamma corrections, contrast stretching, histogram equalization, mosaic-based augmentation, neural style transfer algorithms.

Mosaic – *e*nrich the level of background features in images, localize the target at various scales, four different samples from the training set are randomly combined to form a single image, variations at different scales, increase in batch size without an increase in computational complexity

Neural stye transfer algorithm – improve network's performance under domain variations. To ensure that the content image and style image are combined efficiently, the loss function in Eq. 1 is optimized.

$$\boldsymbol{loss_{total} = \alpha loss_{content} + \beta loss_{style}} \qquad (\mathbf{1})$$

where α and β are the coefficients weighing content loss and style loss respectively

$\boldsymbol{loss_{content}}$ - L2 norm between the content features of the ground truth image and the generated image
$\boldsymbol{loss_{style}}$ - Frobenius norm between the gram matrices of the generated and the ground truth image

Initial dataset – 3100 images of drones and birds

Data augmentation – seven-fold increase (22,000 images)

Data split – train (70%), validation (10%) and test (20%) set

4. METHODS AND EXPERIMENTS

Benchmark Evaluation of YOLO Models

Table 2: Comparison of YOLO models (end of 500 epochs)

Model	Precision	Recall	mAP	Object Loss	Class Loss	fps
YOLOv5	0.8765	0.9032	0.941	0.0252	0.001	123
YOLOv6	0.5231	0.5721	0.855	0.4506	0.646	246
YOLOv7	0.7103	0.6620	0.669	0.0387	0.002	252

- YOLOv5 ***outperforms*** YOLOv6 and YOLOv7 in terms of precision, recall and mAP score – ***comparable*** inference speed (fps)
- YOLOv6 and YOLOv7 present a ***higher inference speed*** measured using a NVIDIA Tesla T4 GPU – ***low classification accuracy***
- YOLOv5 – ***negligible class and object losses*** – quantifies the model's ability to differentiate between the classes
- ***Model training specifications*** – Leaky ReLU activation function for middle layers, Sigmoid activation function for final layers, Adam optimizer, Binary Cross Entropy loss function

Hyper-parameter Evolution Optimization

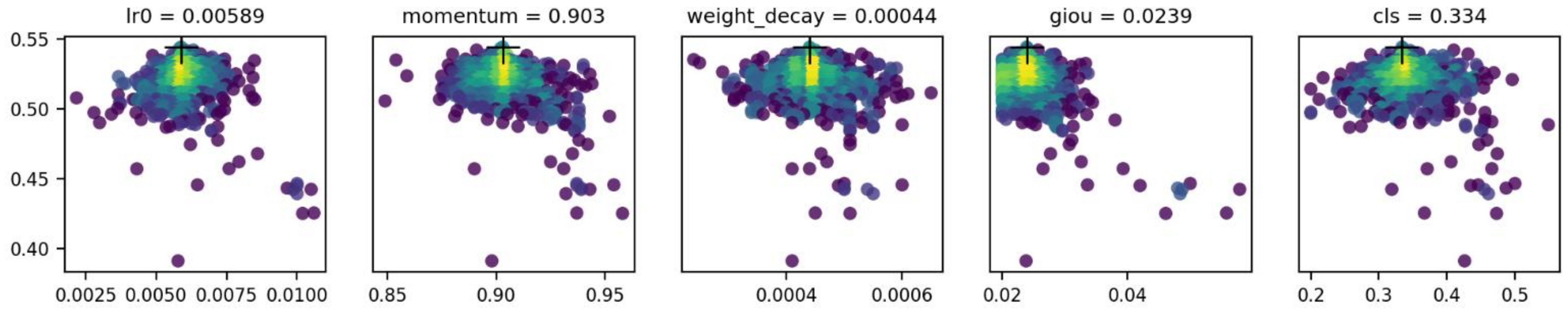


Figure 3: Hyperparameter evolution at the end of 300 generations

- ***Initialization*** – initial population containing N_p vectors created ***with random parameter values***

- ***Mutation*** – For each N_p vector, a ***mutant vector*** is calculated by randomly choosing parameters from the population and each vector's parameter value is computed as a mutation of these randomly chosen parameters. Each parameter p_i of the mutant vector is given by (Eq. 2):
  $$\boldsymbol{p_{i(mutated)} = p_{i(best)} + F \cdot \left(p_{i(r_1)} - p_{i(r_2)}\right)} \qquad \textbf{(2)}$$

- Mutated parameter – variation of p_i of the best vector with the lowest value along with a dot product of the mutation rate F and p_i difference of two vectors randomly chosen, $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$.

- ***Recombination*** – A ***temporary vector*** holds either the ***current vector*** or the ***mutant vector.*** For each of these mutated parameters, a uniform random number R is generated in the (0, 1) interval. If a particular recombination rate is greater than R, the mutant parameter is acceptable else, the parameter of the current vector is used.

- ***Replacement*** – The temporary vector is evaluated for its stability by comparing its function value with the current vector. If it is more stable than the current one, the current vector is substituted for the temporary vector.

4. METHODS AND EXPERIMENTS

ICASSP 2023 4 - 10 JUNE, RHODES ISLAND, GREECE

RAMAIAH Institute of Technology

Sparsity and Model Compression

Model pruning can be viewed as optimizing the pruned network L by minimizing N_p as in Eq. 3

$$\boldsymbol{arg\,\min_{p}\,(L) = N(x;W) - N_p(x;W)} \tag{3}$$

$$\boldsymbol{where\ N_p(x;W) \ = \ P\big(N(x;W)\big)}$$

where N represents the complete neural network with x as the input, L denotes the pruned network with loss in performance given by N_p in comparison to the unpruned network. The pruning function, $P(\cdot)$ represents a compressed network N_p with the pruned weights W_p.

The quantization step adopted can be formulated as in Eq. 4.

$$\boldsymbol{X_q = f(s \times g(X_r) + z)} \tag{4}$$

where s is a scalar, $g(\cdot)$ is the clamp function applied to floating-point values X_r, z is the zero-point to adjust the true zero in asymmetrical conditions and $f(\cdot)$ is the rounding function. The clamping function adopted to quantize the floating point values is as given by Eq. 5.

$$\boldsymbol{clamp(x, \alpha, \beta) = \max(\min(x, \beta), \alpha)} \tag{5}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ represent the bounds for the minimum and maximum values of the parameters respectively.

Deep Sparse – sparsity aware runtime is considered for performance analysis of the models' inference speed

Analysis of Sparse YOLOv5 Models and Network Training

Table 3: Performance analysis of base YOLOv5 models

Model	Layers	Precision	mAP	Total Loss	fps (without compression)	fps (sparse)
YOLOv5n	214	0.921	0.925	0.006	139	606
YOLOv5s	214	0.876	0.941	0.005	123	578
YOLOv5m	291	0.974	0.947	0.005	84	415
YOLOv5l	368	0.978	0.942	0.004	43	203
YOLOv5x	445	0.939	0.939	0.003	22	110

The ***total loss*** is calculated as the sum of box loss, class loss and object loss as given in Eq. 6.

$$loss_{total} = l_{box} + l_{class} + l_{object} \tag{6}$$

The ***box loss*** is shown in Eq. 7.

$$loss_{box} = \lambda_{coordinate} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{i,j}^{o} b_j (2 - w_i h_i) \left[\left(x_i - \hat{x}_i^j\right)^2 + \left(y_i - \hat{y}_i^j\right)^2 + \left(w_i - \hat{w}_i^j\right)^2 + \left(h_i - \hat{h}_i^j\right)^2 \right] \tag{7}$$

where $\lambda_{coordinate}$ is the coefficient of position vector, $I_{i,j}^{o}$ is a variable that holds binary values (0 or 1). If the detected target is inside the anchor box (i, j), then it has a value of 1 else 0.
The penalty function when the network fails to determine the classes accurately is as shown in Eq. 8.

$$loss_{class} = \lambda_{class} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{i,j}^{o} \sum_{c \in class} \overline{p}_i(C) \, log(p_i(C)) \tag{8}$$

λ_{class} is the coefficient of category loss, $p_i(C)$ denotes the true probability outcome of class C and $\overline{p}_i(C)$ is the predicted outcome of class C

The ***object loss*** is computed as given in Eq. 9.

$$loss_{object} = \lambda_{no-o} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{i,j}^{no-o} \left(C_i - \hat{C}_i\right)^2 + \lambda_o \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{i,j}^{o} \left(C_i - \hat{C}_i\right)^2 \tag{9}$$

where λ_{no-o} is the coefficient of object loss.

Non-Maximum Suppression Ensemble Network

Model ensembling – stacking five different versions of YOLOv5 using non-maximum suppression (NMS) algorithm

$$\boldsymbol{s_i} = \begin{cases} \boldsymbol{s_i} & \boldsymbol{IoU(M, b_i) < N_t} \\ \boldsymbol{s_i(1 - IoU(M, b_i))} & \boldsymbol{IoU(M, b_i) \geq N_t} \end{cases} \tag{10}$$

B – list of ***initial detection box proposals*** from each of these models, N_t – ***NMS threshold.*** A proposal from B with the highest confidence score is selected and added to an empty list b_i. ***Intersection over Union*** (***IoU***) is calculated for the selected proposal with every other proposal in the list M. If $\boldsymbol{IoU} > \boldsymbol{N_t}$,this proposal is removed from the list, process is continued until no proposals remain in B. The final confidence score is computed as given in Eq. 10.

Zero Shot Drone Tracking

- ***Instance identification using feature similarity*** across frames
- ***Contrastive Language Image Pre-training*** (***CLIP***) network – detection proposals from the sparse ensemble network
- ***Deep Learning-based Simple Online Realtime Tracking*** (***Deep SORT***) network – tracking instances across frames, assigning unique ***ReID embeddings*** for every distinct object spotted in a frame

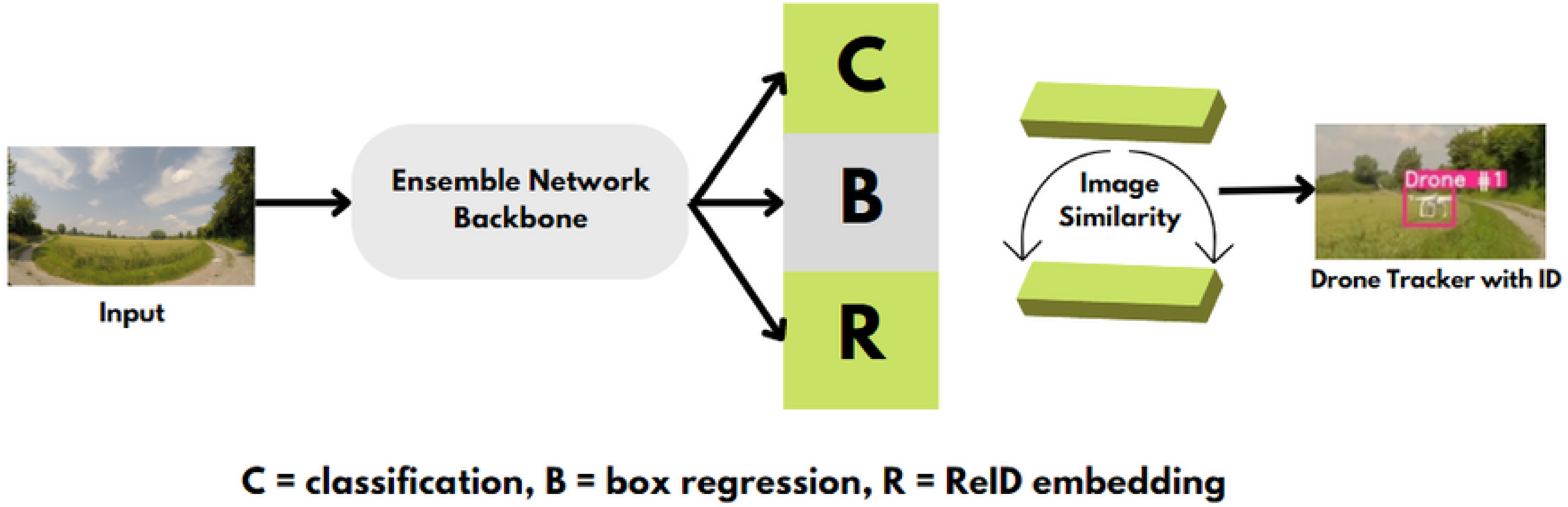


Figure 4: Zero shot drone tracking

5. RESULTS AND DISCUSSION

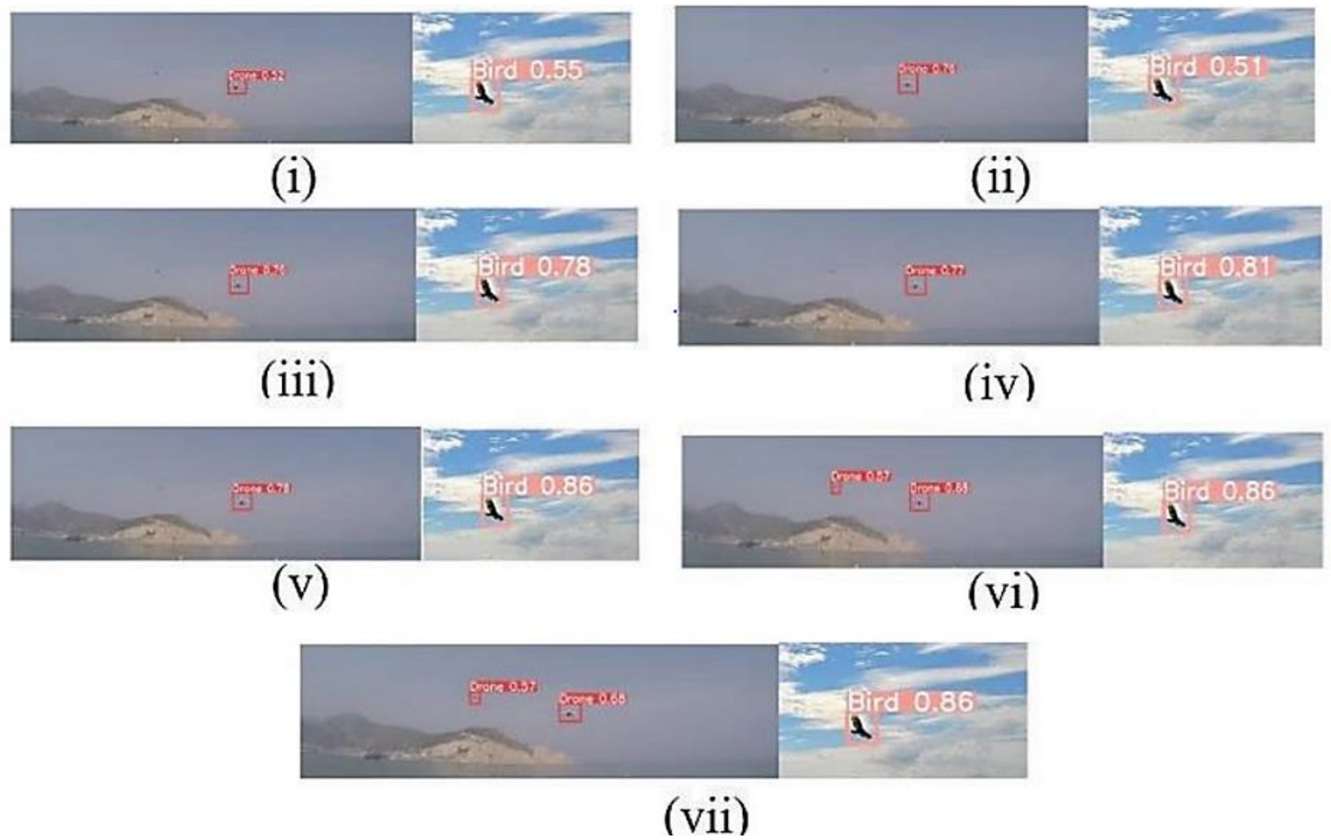


Figure 5: Comparison of individual YOLO models with ensemble network (i) YOLOv5n (0.52) (ii) YOLOv5s (0.76) (iii) YOLOv5m (0.76) (iv) YOLOv5l (0.77) (v) YOLOv5x (0.78) (vi) Ensemble network without compression (0.57,0.68) (vii) SETNET (0.57,0.68)

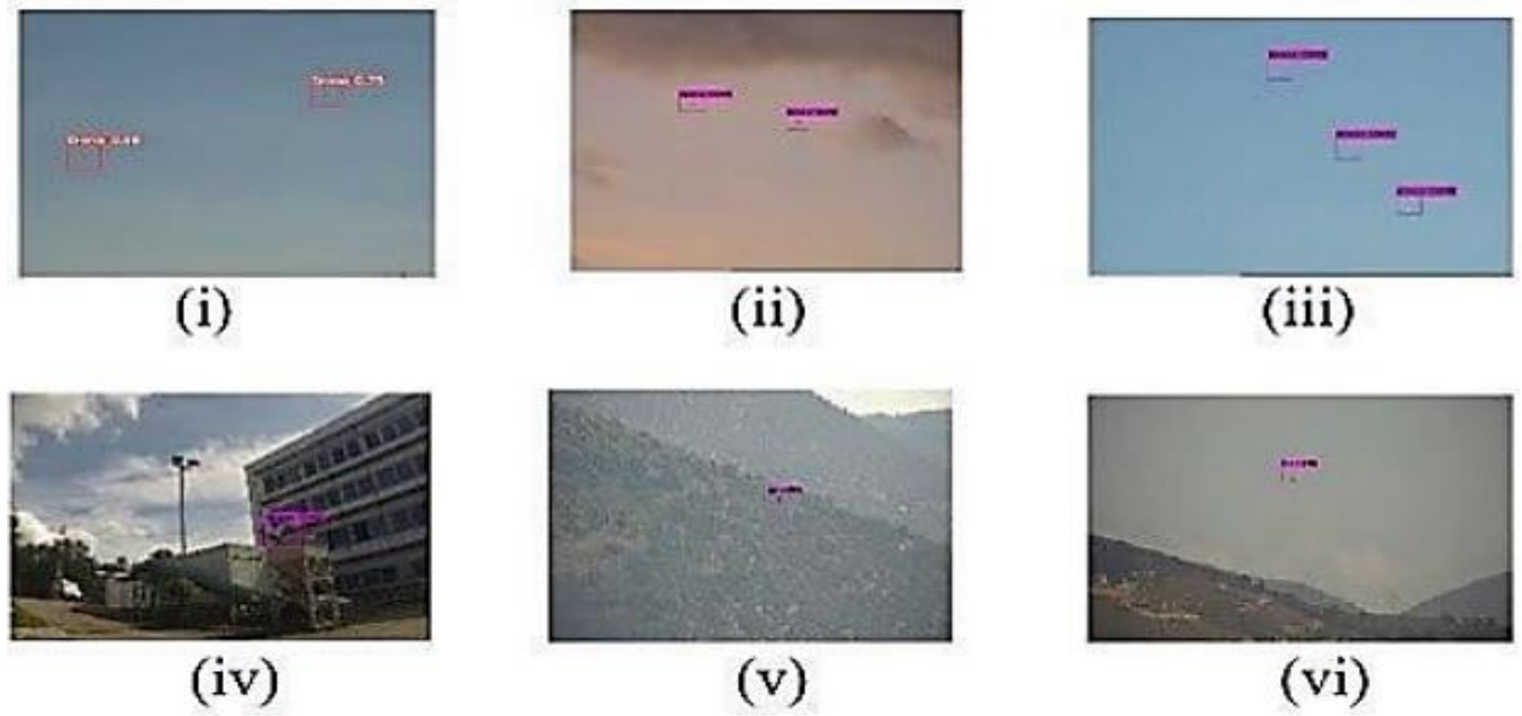


Figure 6: Detection outcomes under unfavourable environmental entities

(i) multiple targets at long range from the viewpoint of the camera source (ii) multiple targets under background irregularities (iii) small targets in clear sky (iv) drone camouflaged by background entities (e.g., building) (v and vi) drone in hilly region with low visibility

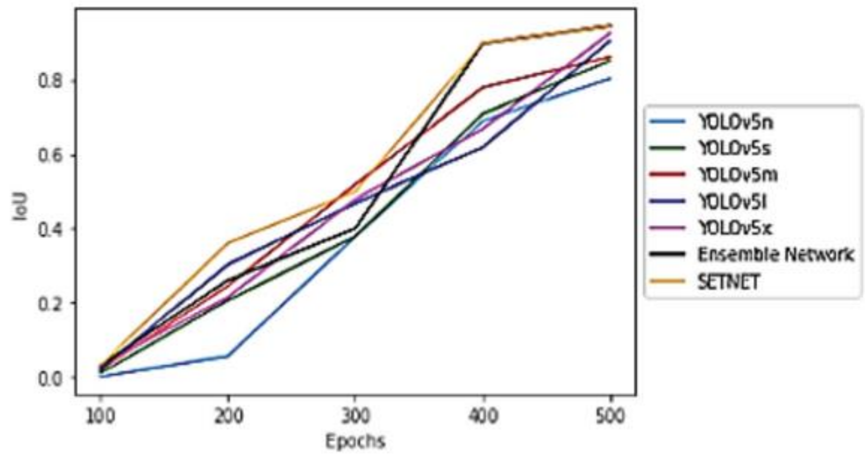
- ***Figure 5 – Two drones*** (*left*) and a ***single bird*** (*right*)
- Confidence of drone localization systematically increases from ***YOLOv5n to YOLOv5x*** (indicated in parentheses)
- Models (i) – (v) individually ***do not*** possess the capability of localizing both the drones that are present in the image on the left
- The ***ensemble network*** (vi) and ***SETNET*** (vii) successfully localize both the target drones although the targets are extremely far-off from the viewpoint of the camera

5. RESULTS AND DISCUSSION

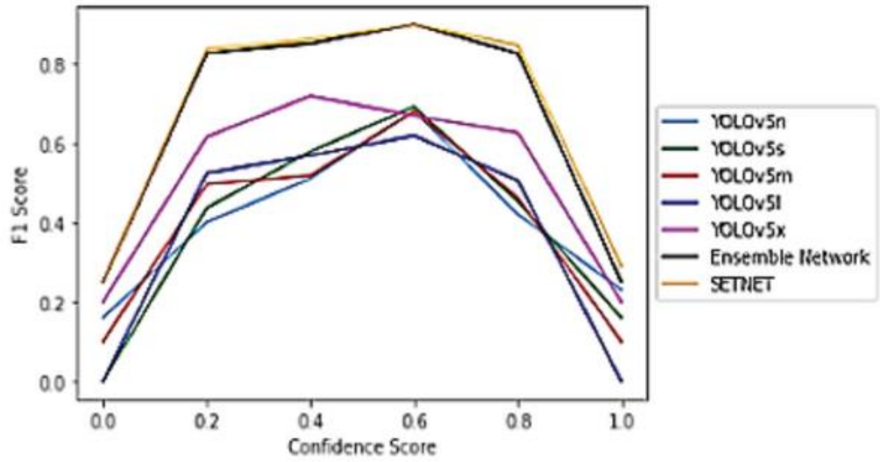
Table 4: Comparison of ensemble network with and without compression

Parameter	Ensemble network (without compression)	SETNET (with compression)
Confidence Score	Drone class (0.68) Bird class (0.86)	Drone class (0.68) Bird class (0.86)
Inference Speed	83 fps	419 fps

- ***Ensemble network vs SETNET***: ***similar*** confidence of detection, SETNET achieves ***five times*** higher inference speed
- Highly suitable for deployment in ***edge computing*** and ***resource constrained environments***



(i)



(ii)

Figure 7: (i) IoU variation with epochs (ii) F1 score variation with confidence score

- SETNET outperforms other models – higher ***Intersection over Union (IoU) score***
- ***Superior*** ability to localize targets accurately that is close to the ground truth
- SETNET achieves higher ***F1 score*** for a given confidence threshold – ***confidence score of 0.7*** yields the maximum ***F1 score of 0.923***

6. CONCLUSION AND FUTURE DIRECTIONS

- SETNET achieves ***robust small target localization*** – extensively evaluated under a variety of environments and scenarios with distortions
- Accounts for ***dynamically changing environment*** and ***topographical conditions***, localizes and tracks small targets under extremely low visibility conditions
- Achieves ***real time drone tracking*** – CLIP-based zero shot tracking framework
- A superior ***five-fold increase*** in inference speed – sparsity in the ensemble network
- Extended to ***infrared images*** – ***image fusion*** approaches for drone localization and tracking (***multimodal learning)***
- Implement – ***real-time surveillance systems*** (***military operations***) in ***resource constrained environments*** and ***edge compute devices***