Onboard Processing of Drone Imagery for Military Vehicles Classification Using Enhanced YOLOv5

Vasavi S *, G. H. Raj, T. Sahithi, and Y. Suhitha

Velagapudi Ramakrishna Siddhartha Engineering College, Andhra Pradesh, India;
Email: 198w1a05h6@vrsiddhartha.ac.in (T.S.), 198w1a05j3@vrsiddhartha.ac.in (Y.S.)
*Correspondence: vasavi_movva@vrsiddhartha.ac.in (V.S.)

***Abstract*—Recently, drones are used in all fields. The video captured by this drone is sent to the terminal for analysis. In terms of speed, performance, and latency, it would be an advantage if the analysis of the image or video is done onboard, the drone, and the result is sent to terminal, this is called onboard processing. For faster recognition speed and higher frame rate, YOLOv5 is used for image detection along with EfficientNet-b0 for classification and de-blurring with DeblurGan v2. A custom dataset of 6999 military vehicle images is created and annotated. This model is loaded in Raspberrypi4 as it is used as a platform to implement real-time image processing applications since their framework can leverage spatial and temporal parallelism. Integrate the Raspberry Pi board into the drone. The classified images are received in a telegram at the terminal. The accuracy of the model is 88%.**

***Keywords*—object detection, military vehicle classification, YOLOv5, drone image, Raspberry Pi 4**

I. Introduction

Drones in the world in every domain and object detection using drones involves the process of the first step is to collect video data from the drone, which typically involves recording footage of the area of interest. Depending on the application, the drone may be equipped with cameras or other sensors that capture visual or other data and the video is sent to the system to perform object detection and classification to get to know the results. The video data is then preprocessed to improve the quality of the data and to make it more suitable for object detection algorithms. This may include tasks such as image enhancement, noise reduction, or data normalization. The core task of object detection involves using machine learning algorithms to detect and classify objects in video data. Several different algorithms can be used for object detection, including deep learning-based approaches such as Convolutional Neural Networks (CNNs) or Region-based Convolutional Neural Networks (R-CNNs) [1] The final step is to visualize the results of the object detection process, typically by overlaying the detected objects on the original video data or by generating heat maps or other visualizations to highlight areas of interest.

If the vehicle detection of video input from the drone is done onboard itself and the result is sent to the ground terminal then it would be, By performing object detection onboard the drone, the processing time can be reduced, which can improve the response time of the system. This can be particularly important in applications where real-time or near-real-time analysis is required, such as in surveillance or monitoring applications, and by processing the video data onboard the drone, the amount of data that needs to be transmitted over the network can be reduced, which can help to reduce network bandwidth requirements and associated costs and also the video data onboard the drone, sensitive information can be kept within the device or system, rather than transmitting it to third-party servers, which can improve privacy and security

By processing the video data onboard, the drone, i.e., the system can continue to function even if the connection to remote servers is lost, ensuring reliable operation. For this, object detection model YOLOv5 along with CNN [2] for classification is used, with the IPWebcam the video is taken from the drone and performed onboard process as explained, and the result is at the system. The vehicles detection model model explained in [3] is considered, and to perform onboard processing and transferring classified images via WI-FI Raspberrypi [4] is considered.

The Custom dataset of 6000 Indian Military vehicles such as Military Tanks, and APCs is collected and annotated using the LabelMe tool, and the object detection model YOLOv5 is trained on the custom dataset. The accuracy of the model is calculated using the IoU.

IoU [5] determines whether the expected bounding box gives us a satisfactory result. Calculates the intersection of the sum of the actual bounding box and the expected bounding box.

The ability of object identification algorithms to recognize an object more than once, in addition to only once, is one of their most typical issues. With non-max

Manuscript received March 23, 2023; revised April 10, 2023; accepted June 6, 2023; published November 16, 2023.

suppression, we select the predictions with the highest confidence level and ignore all other predictions that have more than a threshold of overlap with the selected predictions. In other words, we choose the maximum and attenuate those that are not, hence the name non-maximum attenuation.

II. Literature Review

Because of its potential for usage in applications including traffic control, rescue operations in disasters zone, parking lot management zones, and difficult terrain, this study [1] has highlighted the fact that on-ground automotive recognition from Unmanned Aerial Vehicle (UAV) photos has gained a lot of interest. This article reviews deep learning methods for distinguishing on-ground vehicles from aerial data gathered by UAVs (also known as drones). The works' approach to enhancing accuracy, reducing computer overhead, and achieving their optimization objective was evaluated. Deep learning techniques are used in many ways to improve the microscopic size, dense, and oriented vehicle detection in UAV-based pictures and videos. But when processing UAV images on the ground, latency and security difficulties occur when photos are sent.

Kyrkou and Theocharides [6] focused on the effective categorization of UAV aerial images for use in monitoring and emergency response applications. A comparison of the available methods is done, and a specific Aerial Picture Database for Emergency Response applications is introduced. This investigation leads to the proposal of the Emergency Net architecture, a lightweight convolutional neural network built on around convolutions to handle multi-resolution data and capable of operating well on low-power embedded devices. The advantage of Emergency Net is that it enables us to process even higher-quality photos. Accuracy and performance can be provided for tiny networks where less data is available. But, the Overall classification performance of video streams can be improved.

Lian *et al.* [7] explained wavelet transformations, inverse wavelet transforms, residual depth-wise separable convolution and a DMRFC (dense multi 7 receptive field channel) module. A convolution which is depth-wise separable is created. which, in comparison to the regular convolution, requires fewer model computations and parameters. In contrast to normal convolution and a typical residual block, a depth-wise separable residual convolution is created that enables the transmission of detailed information from several layers. The wavelet transform separates the context and texture information of the image to achieve down sampling. Also, it makes model training easier. Up sampling is achieved through the inverse wavelet transform, reducing the loss of picture data. Their work provides deeper insights into deep learning models for smaller object detection. More Advanced Approaches give much better results.

According to Ref. [8], a CNN is introduced that learns from image-based feature representation at various sizes. The model takes the ground photos, gives different parts of their significance, and determines the landing sites. The appropriate categorization of the ground picture based on its visual content gave support for the model in the findings. They also show that the model may be implemented on a compact computer that is readily integrated into a drone for low computational cost. The algorithms for visual scene interpretation and real-time semantic segmentation may be helpful for low-power drones. The disadvantage of this approach is due to the difference between training loss and validation loss, the model is over-fitted. The training assessment is biased against its data since the validation phase cannot produce the same declining outcomes once the model has "passed" the training set.

Li *et al.* [9] explained a framework for a computationally constrained satellite components recognition model based on YOLOv5 (YSCRM). The difficult multimodal component identification problem is addressed using feature fusion layers and SKNets (selective kernel networks) ,which enhance the model's feature representation and selection capabilities and dramatically increase recognition accuracy. The self-attention mechanism's capacity for prediction is examined at the YOLOv5 Neck's end using the transformer encoder modules. Based on 3D synthetic images data generation and augmentation on consistent cycle adverbial networks were used to increase the quantity and diversity of the data.The five components specified in this approach can all be accurately identified using YSCRM, which has a strong identification capability. The disadvantage of YOLO is that it is unable to recognize and distinguish microscopic items in photographs that appear in groups since each grid can only detect a single item.

According to Ref. [10], it has the first YOLOv5 model's increased capabilities. To identify the critical indications, data must be collected for both the base YOLOv5 model and the enhanced YOLOv5 model. The main improvement is the use of the Conv ELU layer for convolution and the substitution of the SiLU with the ELU activation function. The metrics are recalled, F1-Score, precision, and mAP (0.5). The mAP (0.5) and function loss values were improved by comparing the YOLOv5 Ours model to the original YOLOv5 model. In contrast to YOLOv5's final result, which has a precision of 90.1 and a mAP of 94.6, YOLOv5 Ours has a precision of 90.7 and a mAP of 95.5. Although YOLO's inherent advantage is speed, it has increased prediction accuracy when compared to real-time object detectors and a superior IoU in bounding boxes. The disadvantage of YOLO is that it struggles to recognize and distinguish minute items in photographs if they appear in groups since each grid can only detect one thing at a time.

Chowdhury *et al.* [11] claim that a real-time item-counting technique is based on Raspberry Pi image processing. A technique known as BLOB (Binary Big Object) analysis is used to extract features from a picture that contains objects that need to be discovered. This method employs pixel connectivity as well as other metrics such as area, centroid, and the number of BLOBs present in the test picture to identify the objects or areas

known as BLOBs. The input test image is preprocessed to create a binary image before the BLOB analysis. The number of things discovered is read out or shown via a speaker attached to the device. The Raspberry Pi hardware functions independently once the UBUNTU operating system has been put on a memory card. The Raspberry Pi camera is used as an image acquisition tool to continuously acquire picture frames of interesting items. In contrast to previous computer vision-based technologies, BLOB analysis makes it simple to calculate the characteristics of objects, and the system functions as a standalone device. The major difficulty is to identify or classify items when they are seen in shadow areas.

According to Yuan *et al.* [12], the Raspberry Pi is always looking for information from PIR sensors or calling bells. Moreover, this indicator causes a particular alert to be sent. The algorithm states that if a visitor enter and rings the doorbell, both interruptions occurred, which is a typical circumstance and a sign that a visitor has arrived, but if just one disruption occurs, such as human motion, it's a sign that a thief or robber may have arrived. A single interruption of the calling bell indicates an issue with the PIR sensor. After the system experienced both interruptions, the Raspberry Pi uses its Wi-Fi camera to take a picture of the user. Here, a Wi-Fi network is used to link the camera to the Raspberry Pi [26]. The authorized person receives the E-mail about the visitor and picture as soon as the PIR sensor is disrupted. But, if the intruder moves at faster rate, then detection of person becomes difficult and at nights it is not possible.

III. Materials and Methods

An outline of the proposed system architecture, methodology, algorithms, and dataset for the designed system are presented here.

- Model a. Segmentation model for vehicles
- Model b: Enhanced YOLOv5 model

A. Architecture

Fig. 1 presents the proposed system model for the segmentation of vehicles [13].

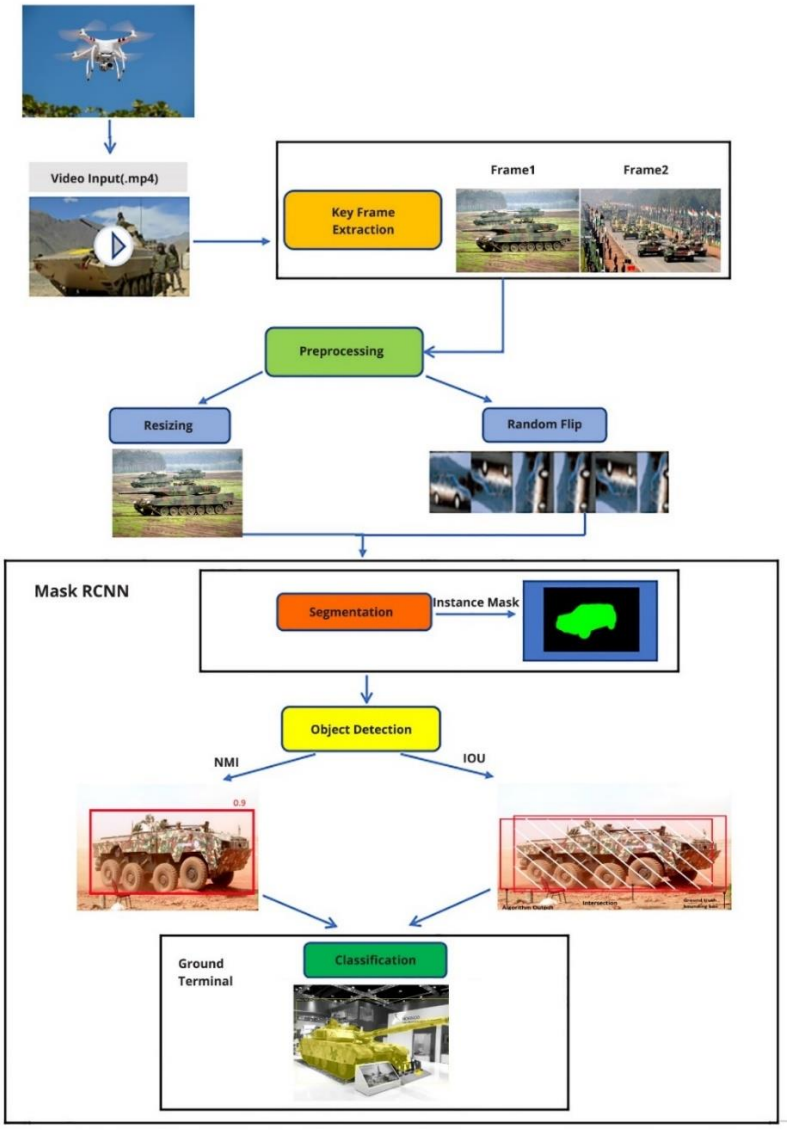

Figure 1. Process flow diagram for model 1.

A network architecture design is comprised of live video from drone using IP web cam through WI-FI. Fig. 1 shows the proposed architecture. In the proposed architecture, the image of 1024×1024 is given to DeblurGanv2 for image deblurring. As per research, out of various object detection models YOLO, R-CNN, Faster R-CNN [14–17]. YOLOv5 is the one with a good quality and accurate outcome. Advantages of YOLO architecture compared to other object detection models are efficient use of data, Preserving high-resolution features using skip connections, Flexible, Highly accurate. To the existing YOLO model, a CNN model of 50 layers is added for classifying the military vehicles. In YOLOv5, the initial weights are typically randomly initialized using a normal distribution, normal distributed random values with a mean of 0 and a standard deviation of 0.01. Table I presents the details of the modified YOLOv5 model proposed in this study (see Fig. 2).

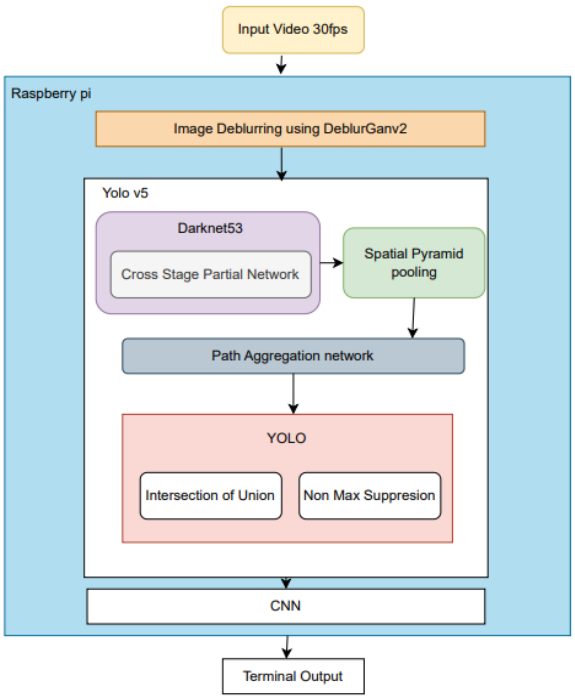

Figure 2. Proposed architecture for model 2.

B. Methodology

In this section, we'll talk about the various parts of the project that were put together. The modules included are Dataset Collection, Data preprocessing, Training YOLOv5 model, Deployment into Raspberry Pi, Result sent to telegram. Fig. 3 represents the training process flow of the system.

As shown in Fig. 3, in the training phase, the data is collected initially and labelling of images is performed using label me tool. After that, data preprocessing is performed. In that, to remove the image blur, image de-blurring technique DeblurGan v2 is used. The YOLOv5 Model is trained using the dataset collected and model is deployed into Raspberry Pi. Fig. 4 represents the testing process flow of the system.

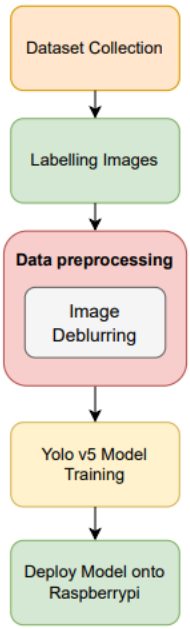

Figure 3. Training flowchart.

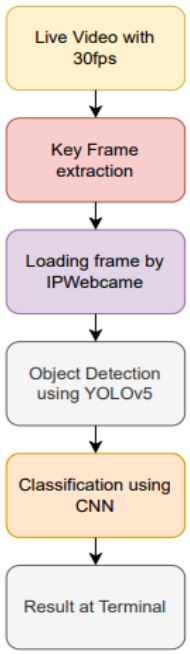

Figure 4. Testing flowchart.

As mentioned in [18], extraction of features is done by employing a smaller 3×3 convolutional layers. To ensure real-time operation on embedded hardware as specified in [19] only the features that require a relatively small number of non-sophisticated computations are selected. For attaining dimensionality reduction, two as step size is used by our enhanced YOLOv5 model whereas convolutional layer with one as the step size is used by YOLOv5 [20]. The convolutional layer structure employs the same structure as YOLOv5, i.e., Convolution2D + Batch Normalization + Leaky Relu. Darknet 53 is used in YOLOV5 and in enhanced YOLOv5 model Darknet53+EfficientNet 227 [21] layers are used.

TABLE I. MODIFIED YOLO MODEL DETAILS

Parameters	Existing Model [14]	Modified Model
Channels in input image	3	3
Shape of input image	(640,640,1)	(1024,1024,3)
Strides	1	2
Input Kernel size	3×3	3×3
Parameters	7.31M	21.7M
Pooling type	Max Pooling	Max Pooling
Size of Max Pooling at every layer	2×2	2×2
No. of Layers	Darknet 53	Darknet 53+EfficientNet-b0 227
Channels in output image	1	1

1) Data collection

Data preparation is the initial step of any DL project which involves gathering raw data to train the model. Datasets need to be prepared well before using them to increase model performance. The object detection model has been trained to detect APCs and military tanks. However, an open dataset containing various classes of military vehicles that is suitable for object detection is not available anywhere. The images of military vehicles belonging to India shave been taken from Kaggle [22]. Additionally, the images for APCs have been extracted from ImageNet Dataset. ImageNet [23] is a large image database which consists over 14 million images belonging to different categories. A total of 6000 images containing Military Vehicles are collected as the dataset. The videos for testing the model have been taking from GitHub. The original base image, hence the labels were resized to the resolution of the base image and moved to the next module.

2) Pre-processing

The images taken from the drone consists of motion blur associated with them, in order to remove the motion blur associated with frame, deblurgan v2 is used.

When an image is affected by blur, it can appear blurry, distorted. De-blurring [24] from the image can improve its overall appearance, but it may not necessarily restore all the lost information.

3) Military vehicle detection and classification

Military Vehicle detection is performed using the modified YOLOv5 model. The input frame is extracted and after performing data preprocessing the frame is sent into YOLO to perform bounding box prediction. Initially, the model was trained with 53 layers. But while classification of the vehicles requires more speed and less latency, hence along with YOLO, an EfficientNet-b0 model of 227 layers is used to classify the military vehicles.

C. Algorithms

1) Key frame extraction

The Algorithm of Key Frame Extraction, i.e., dynamic Clustering [24] is as of follows:

Step 1: Load the Video from which frames are to be extracted.

Step 2: All three channels (RGB), which each have 6 bins, are used to create the colour histogram for each of the 3×3 blocks, or 9 blocks of frames in the video.

Step 3: In order to build a feature-frame matrix with dimensions (1832, 1944) for the full film, the nine histograms were concatenated to create 1944-dimensional feature vectors for each frame. There were 1832 frames in all.

Step 4: Using SVD dimension of matrix is reduced to (1832,63)

Step 5: The clusters of successive frames are examined using cosine similarity to determine whether or not the new frame is comparable to the previous cluster.

Step 6: The frames in the sparse clusters were disregarded since they are regarded as transitions between shoots. A shot is made up of a dense cluster of frames, with the final frame added acting as a key frame.

2) DeblurGanV2

The Algorithm for image deblurring using DeblurGanv2 [25] compared to that of image deblurring [26]. DeblurGanv2 algorithm is described below:

Step 1: The feature extraction network is typically based on a pre-trained VGG-19, which is used to extract high-level features from the input image. The deblurring network is responsible for generating a deblurred image from the extracted features as given in Eq. (1).

$$\mathbf{G}(\mathbf{x}) = \mathbf{F}(\mathbf{x}) + \mathbf{H}(\mathbf{x}) \quad (1)$$

Step 2: The deblurring network H(x) can be formulated with Eq. (2).

$$\mathbf{H}(\boldsymbol{\varkappa}) = \mathbf{tan\,h}(\mathbf{wo} * \mathbf{re\,lu}(\mathbf{w_1} \times \mathbf{x} + \mathbf{b_1}) + \mathbf{b_0}) \quad (2)$$

where W1 and W0 are trainable convolutional filters, b1 and b0 are trainable bias terms, and relu() and tanh() are activation functions. The output of the deblurring network H(x) is a deblurred image.

Step 3: The discriminator network takes both the input blurred image and the output deblurred image from the generator as inputs and outputs a binary classification score indicating whether the deblurred image is real or fake.

Step 4: The discriminator network can be formulated with Eq. (3)

$$(\mathbf{x}, \mathbf{y}) = \mathbf{s}\big(\mathbf{C}(\mathbf{x}, \mathbf{y})\big) \quad (3)$$

where *x* is the input blurred image, *y* is the output deblurred image from the generator, *C* (*x*, *y*) is the concatenation of *x* and *y* along the channel dimension, and S() is the sigmoid activation function.

Step 5: Define a loss function that captures the difference between the generated deblurred image and the ground truth sharp image, as well as the ability of the discriminator to distinguish between real and fake images. The DeblurGANv2 algorithm uses a combination of adversarial loss, perceptual loss, and Mean Square Error (MSE) loss.

The adversarial loss can be formulated with Eq. (4):

$$\boldsymbol{L_{adv} = -Log(D(x, G(x))} \tag{4}$$

Step 6: Training: Use the dataset of sharp and blurry image pairs to train the DeblurGANv2 model. The generator and discriminator networks are trained alternately, with the generator attempting to produce realistically blurred images that deceive the discriminator and the discriminator attempting to differentiate between genuine and created deblurred images.

3) Segmentation

Input: Images of the height 600 and width 800 is given as input.

Output: Binary Segmented mask is generated.

Step 1: Mask R-CNN comprises of a FPN backbone that predicts ROI with a Faster R-CNN branch that perform classification and bounding box regression in parallel to a mask layer that generates masks.

Step 2: The mask branch is a FCN applied to each ROI, predicting a binary segmentation mask in a pixel-to-pixel manner.

The multi-task loss in each sampled ROI is as follows using Eq. (5):

$$L = L_{cls} + L_{box} + L_{mask} \tag{5}$$

4) YOLOv5

The Algorithm for Predicting Bounding boxes is described below:

Step 1: The input image is resized to the desired input size, which is usually a square image.

Step 2: The input image is passed to a modified version of the CSPNet (Cross Stage Partial Network) as the backbone network.

Step 3: The features extracted using the SPP (Spatial Pyramid Pooling) and PAN (Path Aggregation Network) modules .

Step4: The refined features are passed through a set of convolutional layers and predict the boxes bounding the objects in the image.

Step 5: Post processing: The predicted bounding boxes are post processed to remove duplicates and overlapping boxes. Non-Maximum Suppression (NMS) is used to remove redundant boxes and keep only the most confident predictions.

Anchor boxes: Anchor boxes are used to define the possible locations and scales of objects in the image. YOLOv5 uses k-means clustering to find the optimal anchor box sizes based on the ground truth bounding boxes

The formula for calculating the anchor box width and height is shown in Eq. (5a)–(5b):

$$w = \sqrt{\frac{\text{box\_width} \times \text{box\_height}}{k}} \tag{5a}$$

$$w = \sqrt{(\text{box}_{\text{width}} \times \text{box}_{\text{height}}) \times k} \tag{5b}$$

Step 6: Output the image with Predicted bounding box.

Intersection over Union (IoU): IoU is a measure of the overlap between two bounding boxes as given in Eq. (6).

$$\text{IoU} = \text{Area of intersection} \,/\, \text{Area of union} \tag{6}$$

5) CNN algorithm

The Algorithm for Classification of military vehicles described below:

Step 1: Initially read the input image 1and resizes the image, convert it to a tensor, and normalize the pixel values.

Step 2: Initialize the weights and biases of the network: Randomly initialize the values of the weights and biases, or use pre-trained weights and biases if available.

Step 3: Activations of each layer are computed using forward propagation, passing the outputs of one layer as inputs to the next layer.

Step 4: Compare the predicted class probabilities with the true class labels.

D. Evaluation Metrics

The results of the detection task were evaluated for network quality using the metrics proposed. Here are some of the measurements that were taken:

You can evaluate a model's efficacy using a number of different metrics, including precision (P), recall (R), crossroads over union (IoU, Jaccard Index), and F1 score (Dice coefficient).

As such, the corresponding equations are Eqs. (7)–(10). The number of pixels that were properly detected as cars is known as True Positive (TP), whereas the number of pixels that were mistakenly classified as absences of vehicles is known as False Positive (FP). False Negative (FN) indicates how many pixels were mistakenly classified as automobiles whereas True Negative (TN) indicates how many pixels were accurately detected as the absence of a vehicle. Findings were presented using the average IoU for both the structure and the environment groups.

$$Precision = \frac{TP}{TP+FP} \tag{7}$$

$$Recall = \frac{TP}{TP+FN} \tag{8}$$

$$IOU = \frac{TP}{TP+FP+FN} \tag{9}$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{(Precision+Recall)} \tag{10}$$

IV. Result and Discussion

The results are obtained through the successful execution of the proposed system of Model 2. Fig. 5

shows the classification report for 6 military vehicle classes. It represents the frame extracted from the live video .

Classification Report

	precision	recall	f1-score	support
military tank	0.84	0.85	0.84	78
civilian car	0.88	0.89	0.89	103
military helicopter	0.85	0.86	0.85	105
military aircraft	0.97	0.89	0.93	139
military truck	0.91	0.89	0.90	166
civilian aircraft	0.83	0.90	0.86	127
accuracy			0.88	718
macro avg	0.88	0.88	0.88	718
weighted avg	0.89	0.88	0.88	718

Figure 5. Classification report.

The Accuracy of the model 2 is 88%. The training times and prediction times for proposed YOLOv5 architectures are 12.420 s and 92.3 s.

Fig. 6 represents deblurred image after performing the image de-blurring using Deblurganv2. As to remove the motion blur associated with the extracted frame.

Figure 6. Key frame extracted from the live video.

Fig. 7 represents the classified military vehicle using EfficientNet-b0 model that is military truck. Fig. 8 presents the Model Accuracy Graph. Fig. 9 presents the Model Loss Graph.

Figure 7. De blurred image of military truck.

Figure 8. Classified military vehicle.

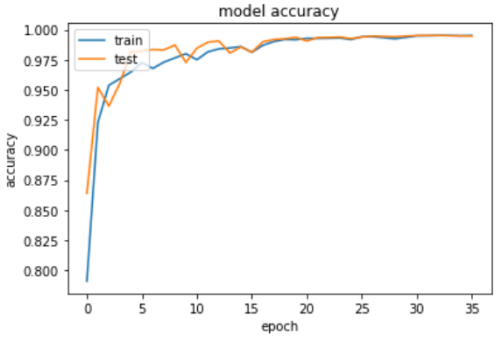

Figure 9. Model accuracy graph.

As we can see in the above graph, the accuracy of the model has been improved with the increase in the number of epochs (see Fig. 10).

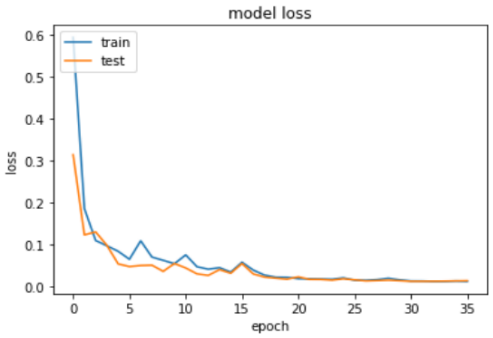

Figure 10. Model loss graph.

From the above graph, it is clear that the loss kept on minimizing with the epochs which started giving accurate results. Fig. 11 represents the Roc curve for multiclass classification.

ROC curve is plotted using macro averaging approach in which, calculates the true positive rate and false positive rate for each class separately and then take the average across all classes to get a single point on the ROC curve,. The classes used are military tank, civilian car, military helicopter, military aircraft, military tank, civilian aircraft. As specified in [27–29] the model's true positive rate has improved from 0.4 to 1.0 over the iterations and Area Under Curve (AUC) obtained is 0.97. Table II presents the comparison with various other models.

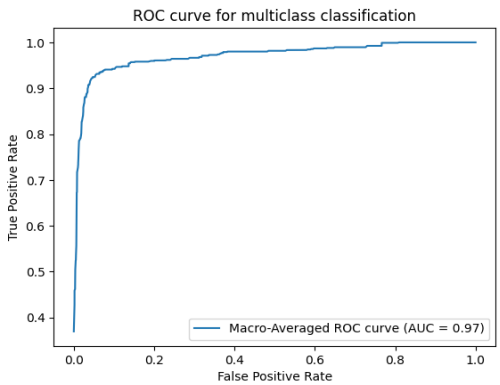

Figure 11. ROC curve.

TABLE II. COMPARISON WITH OTHER MODELS

Study	Methodology	Accuracy Test	Accuracy
Kyrkou and Theocharides [6]	Atrous Convolutional Feature Fusion	Mean IoU	0.85
Li *et al*. [10]	Improved YOLOv5 using Attention Mechanism	Mean IoU	0.82
Jung and Choi [1]	Improved YOLOv5: Efficient Object Detection Using Drone Images	Mean IoU	0.92
Chen *et al*. [18]	YOLOv5 model for vehicle detection.	IoU	0.90
Proposed Work	Modified YOLOv5 along with CNN	Mean IoU	0.88

Lightweight architectures bring benefits such as faster inference speed, lower resource requirements, reduced network bandwidth, improved portability, and deployment flexibility to real-time military vehicle detection. Their applicability and performance make them well-suited for deployment on devices with limited computational power, memory, or energy resources. This extends the applicability of this model to scenarios where more heavyweight models may not be feasible or performant, such as embedded systems, IoT devices, or edge computing environments, enabling efficient and effective detection of military vehicles in various operational scenarios.

V. CONCLUSION

In this study, YOLOv5-based military vehicle detection along with Efficient net for classification is used. A dataset for this project work is prepared with 6999 Indian military vehicles are considered. The model has been trained on preprocessed data which includes the deblurring of images because of motion blur, which helps in improving the accuracy of the model. Model(a) is the segmentation model which provides less accuracy, i.e., Mean IoU of 0.87 and Model(b) is a classification model using enhanced YOLOv5 with EfficientNet [30] gave a accuracy of 0.88 as it consists of less computation and more accuracy, Model(b) is more preferred.

The research provides a practical basis to perform vehicle detection and classification on Raspberry Pi [31] integrated to drone and send result to ground terminal. This study helps the military in identifying the military vehicles from the surveillance. It helps in performing Real-time and low-latency processing.

Our Future work mainly consists of improving the model by using a large dataset and also to use microprocessor such as FPGA [32] for better processing. Furthermore to detect vehicles during low-light environments, also to detect from higher altitude [33] and to improve the model accuracy.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

S. Vasavi for Conceptualization, Methodology, Writing—Original draft preparation, Validation, Reviewing and Editing; G. H. Raj for Software, Writing—Original draft preparation, Visualization, Validation; T. Sahithi and Y. Suhitha for Software, Writing—Original draft preparation, Reviewing and Editing. All authors had approved the final version.

REFERENCES

[1] S. Srivastava, S. Narayan, and S. Mittal, "A survey of deep learning techniques for vehicle detection from UAV images," *Journal of Systems Architecture*, vol. 117, 102152, 2021.

[2] K. O. Shea1 and R. Nash, "An introduction to convolutional neural networks," arXiv preprint, arXiv:1511.08458, pp. 1–11, 2015.

[3] J. Lu *et al.*, "A vehicle detection method for aerial image based on YOLO," *Journal of Computer and Communications*, vol. 6, pp. 98–107, 2018.

[4] R. Kashaboina and R. Velmani, "Bluetooth and GSM based smart security system using Raspberry Pi," in *Proc. IOP Conference Series: Materials Science and Engineering*, vol. 981, 2020.

[5] Intersection over Union (IoU) for object detection. [Online]. Available: https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

[6] C. Kyrkou and T. Theocharides, "EmergencyNet: Efficient aerial image classification for drone-based emergency monitoring using atrous convolutional feature fusion," *Institute of Electrical and Electronics Engineers*, vol. 13, pp. 1687–1699, 2020

[7] Z. Lian, H. Wang, and Q. Zhang, "An image deblurring method using improved U-Net model," *Mobile Information Systems*, vol. 2022, pp. 1–11, 2022.

[8] O. Bektash, J. J. Naundrup, and L. A. C. Harbo, "Analyzing visual imagery for emergency drone landing on unknown environments," *International Journal of Micro Air Vehicles*, vol. 14, 2022.

[9] C. Li, G. Zhao, D. Gu, and Z. Wang, "Improved lightweight YOLOv5 using attention mechanism for satellite components recognition," *IEEE Sensors Journal*, vol. 23, pp. 514–526, 2023

[10] H. K. Jung and G. S. Choi, "Improved YOLOv5: Efficient object detection using drone images under various conditions," *Applied Sciences*, vol. 12, no. 14, 7255, 2022.

[11] M. J. T. Pramod, S. A. K. Jilani, and S. J. Hussain, "Real time object counting using Raspberry Pi," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, 2023.

[12] N. Chowdhur, S. Nooman, and S. Sarker, "Access control of door and home security by Raspberry Pi through internet," *International Journal of Scientific and Engineering Research*, vol. 4, no. 2229–5518, 2013.

[13] S. Vasavi, D. S. Soumya, C. Aishwarya, and W. F. Fuentes, "FPGA based military vehicle classification from drone-based video using mask RCNN," *Communicated to IEEE Transaction for Image Processing*, vol. 5, 2022.

[14] R. Chatterjee, A. Chatterjee, and S. H. Islam, "Deep learning techniques for observing the impact of the global warming from satellite images of water-bodies," *Multimed Tools Appl.*, vol. 81, pp. 6115–6130, 2022.

[15] D. Cao, Z. Chen, and L. Gao, "An improved object detection algorithm based on multi-scaled and deformable convolutional

neural networks," *Hum. Cent. Comput. Inf. Sci*., vol. 10, no. 14, 2020.
[16] P. Malhotra and E. Garg, "Object detection techniques: A comparison," in *Proc. 2020 7th International Conference on Smart Structures and Systems (ICSSS)*, Chennai, India, 2020, pp. 1–4.
[17] P. Gupta, B. Pareek, G. Singal *et al*., "Edge device based military vehicle detection and classification from UAV," *Multimed Tools Appl*, vol. 81, pp. 19813–19834, 2022.
[18] Z. Chen, L. Cao, Q. Wang, "YOLOv5-based vehicle detection method for high-resolution UAV images," *Mobile Information Systems*, vol. 1, pp. 1–11, 2020.
[19] Live video. [Online]. Available: https://drive.google.com/file/d/114FmpakvAqs3gPZIC0gIWMUMEGUUpHST
[20] S. Astapov, A. Riid, and J. S. Preden, "Military vehicle acoustic pattern identification by distributed ground sensors," in *Proc. 2016 15th Biennial Baltic Electronics Conference (BEC)*, 2016, pp. 167–170.
[21] M. Prashnani and R. S. Chekuri, "Identification of military vehicles in hyper spectral imagery through spatio-spectral filtering," in *Proc. 2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013)*, 2013, pp. 527–532.
[22] Military Tanks Dataset. [Online]. Available: https://www.kaggle.com/antoreepjana/military-tanks-dataset
[23] Image Net Datasets Downloader Public. [Online]. Available: https://github.com/mf1024/ImageNet-datasets-downloader
[24] Bouchachia, "Dynamic clustering," *Evolving Systems*, vol. 3, no. 3, pp. 133–134, 2012.
[25] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better," arXiv preprint, arXiv:1908.03826, 2019.
[26] L. Yuan, J. Sun, L. Quan, and H. Y. Shum, "Image deblurring with blurred/noisy image pairs," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 1, 2007.
[27] G. Altan, "DeepOCT: An explainable deep learning architecture to analyze macular edema on OCT images," *Engineering Science and Technology, an International Journal*, vol. 34, 101091, 2022
[28] G. Altan, Y. Kutlu, and N. Allahverdi, "Deep learning on computerized analysis of chronic obstructive pulmonary disease," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 5, pp. 1344–1350, May 2020.
[29] G. Atlan, "Deep learning-based mammogram classification for breast cancer," *International Journal of Intelligent Systems and Applications in Engineering (IJISAE)*, vol. 8, no. 4, pp. 171–176, 2020.
[30] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. International Conference on Machine Learning*, 2019, pp. 6105–6114.
[31] H. D. Ghael, L. Solanki, and G. Sahu, "A review paper on Raspberry Pi and its applications," *International Journal of Advances in Engineering and Management*, vol. 2, pp. 225–227, 2020.
[32] S. P. Kaarmukilan, S. Poddar, and A. K. Thomas, "FPGA based deep learning models for object detection and recognition comparison of object detection," in *Proc. the Fourth International Conference on Computing Methodologies and Communication (ICCMC 2020)*, 2020, pp. 1–13.
[33] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, 217, 2021.