Military Aircraft Detection and Classification

Vidhi Kokel
Department of Electrical and Computer Engineering
Queen's University
vidhi.kokel@queensu.ca

Abstract

In the current times of technological evolution in the Computer Vision, Machine Learning and Deep Learning domains, applying these advancements in the domain of defence of a country becomes inevitable. This paper attempts to detect the military aircraft from the provided image and classify its type from the 39 different categories. This could be used by the armed forces of the government of various countries to estimate the intensity of the attack they might be under from the detected type of aircraft. This paper uses YOLOv5 model for the detection as well as classification and also presents the results after performing various experiments for the same.

1. Introduction

Conventionally various tools like manual binoculars were utilised to detect the size and shape of an aircraft along with its sound to identify its type. But many a times the complex nature of backgrounds in battlefield make these techniques less effective. Hence, for situations like these various works have been carried out to automate this process making it more flexible and portable for it to be applicable in a military setting.

For performing this task of aircraft identification and detection, multiple techniques are utilised for the feature extraction and then passed on to the deep learning or machine learning based models for their respective classification. Because of its obvious and critical application in the military, various techniques have been constantly tried, tested, updated and replaced in order to identify an efficient technique for the same. Compared to other detection models, YOLO has a better generalization effect in some applications. Moreover, it is capable of providing results in real-time. Hence, this study is carried out by implementing the latest version of YOLO i.e. YOLOv5.[1]

The rest of this paper is coordinated as follows. In Section 2, we briefly discuss about the motivation and the related works that have been carried out for similar use cases. In Section 3, we talk about the dataset, methodology as well as the experiments performed and report their respective results in section 4. We then interpret from the methodologies and results and provide a glimpse of future work in section 5 along with a conclusion in section 6 that summarises all the work. Lastly, some code citations are reported in the section 7.

2. Related Work

With incremental improvements in the target detection technologies, there have been many techniques for detection and recognition of objects which could be chosen according to the areas they are applied to. But from them, majorly two techniques stand out in the terms of efficiency, speed and overall performance. They are region based object recognition which consists of models like region convolutional neural network (RCNN), Fast-RCNN etc. and object recognition based on regression which consists of models like YOLO, SSD (single shot multibox detector) etc. Following is a discussion of some such works that employ these recognition techniques and evaluate their respective results.

Lu Chen et al. and their team propose an architecture where they introduce an inception module and a multi-scale prediction in the YOLOv3 architecture to improve the results given by YOLOv3 model alone for aircraft identification. They have used pre-trained YOLOv3 model along with reclustering of the dataset. All these contributed towards improved performance metrics like recall, F1 score etc. Additionally they performed some experiments for this trained model to detect other objects to prove the generality and wide applicability of the proposed model, which also showcases promising results.[1]

Hai Jun Rong et al. and their team formulate a scheme to recognize different types of aircrafts. In this scheme, they use three different modular neural networks as classifiers to which they provide a different moment invariant respectively as input features. The invariant moments used for this scheme are Hu, Zernike and Wavelet. Later, to get the final result they combine the results from the outputs of these modular neural networks. Finally, they compare the results

for this modular extreme learning model with the individual invariant moment extreme learning models and modular networks with other classification algorithms. All the comparisons point out towards the proposed technique being superior of them all.[2]

Roopa et al. and their team discusses an identification system for aircrafts using invariant moment technique and discrete cosine transform method and compare their results respectively. They deduct that discrete cosine transform method yields better results for the provided static set of images. Additionally, they suggest to use multiple view aircraft images for the same class in order to further improve the performance.[3]

Syed Faisal Ali et al. and their team employ a conventional feature extraction technique in the Video Automatic Target Recognition Systems (VATRs) to make it intelligent for aircraft recognition. The technique they use here is termed as WEFT, where they identify the Wings, Engine, Fuselage and Tail of the aircraft manually and then pass it on to the VATRs for further processing and identification. Employing this into an VATRs makes it more applicable because of the flexibility and portability of a VATRs.[4]

Thus, after referring to the above works, YOLOv5 being the most efficient, reliable and fast deep learning model for real-time object recognition and detection, it has been chosen as the primary model for this study.

3. Methodology

Overall methodology of this study consists of dataset description and splitting, deep learning based model configuration and lastly its training with different combinations of parameters to record and interpret the obtained results for the various experiments performed. Dataset sub-section describes the dataset chosen for this study along with reporting the splitting logic chosen for the training and validation sets of data. Model Configuration sub-section discusses the general architecture of YOLOv5 model and details the dimensions of the different layers of the implemented YOLOv5 model. Lastly, the experiments sub-section discusses about the undertaken investigations performed.

3.1. Dataset

The dataset chosen for this study is Military Aircraft Detection Dataset which is publicly available at Kaggle at `https://www.kaggle.com/datasets/a2015003713/militaryaircraftdetectiondataset`. This dataset consists of military aircrafts of 39 different categories(A10, A400M, AG600, B1, B2, B52 Be200, C130, C17, C5, E2, EF2000, F117, F14, F15, F16, F18, F22, F35, F4, J20, JAS39, MQ9, Mig31, Mirage2000, RQ4, Rafale, SR71, Su34, Su57, Tornado, Tu160, Tu95, U2, US2, V22, Vulcan, XB70, YF23), their respective labels in PASCAL VOC format (xmin, ymin, xmax, ymax) along with their respective annotated images. Figure 1 shows the sample images from the dataset, where it can be seen that these images have different varieties like multiple backgrounds, presence of more than 1 aircrafts in asingle image, presence of other objects in the image etc.

For this study we utilise the raw images without annotations along with the labels provided in the .csv format. Here, in order for the YOLOv5 to work as expected, we convert the format of labels from PASCAL VOC to the format compatible with YOLOv5 which is a text file containing a row of labels (class of object, xcenter of the bounding box of object, ycenter of the bounding box of object, width of the bounding box, height of the bounding box) for each object in the image. Moreover, we further normalize the numeric values in the labels by dividing the X-axis related values with the width of the bounding box and Y-axis related values with the height of the bound box respectively. Lastly, we split the dataset into two parts i.e. training and validation using 70% of the provided dataset for training and 30% of it for the validation.

3.2. Model configuration

Here, in this study we utilise the YOLOv5 deep learning model for both detection and classification of the military aircrafts present in the provided images because of it's high performance and that too using very few resources. The basic architecture of YOLOv5 model is represented in Figure 2. The Backbone of the model is responsible for pre-training, the neck for collection of feature maps and the head is responsible for detecting the bounding boxes and predicting the class for the object/s in the image.

There are various versions of YOLOv5 according to the hardware constraints of the devices on which the model will be executed. To make the implemented model of this study more ubiquitous, easy to use and to obtain the results as fast as possible i.e. in real-time, we chose YOLOv5s version. The detailed layers and their respective dimensions of the model implemented here can be seen in Figure 3. For this study, we have used the pre-trained weights to train the model.

3.3. Experiments

After pre-processing and splitting the dataset, here the model is trained using the obtained training images in different batches for numerous epochs. After the training is completed successfully, the model validates its training by classifying the classes for the validation imageset. Lastly, all the performance metrics for the deep learning model are evaluated by the model and reported to the user.

In this study, we have performed two experiments. One where we provide the whole dataset to the model and another where we are randomly choosing 500 images from all

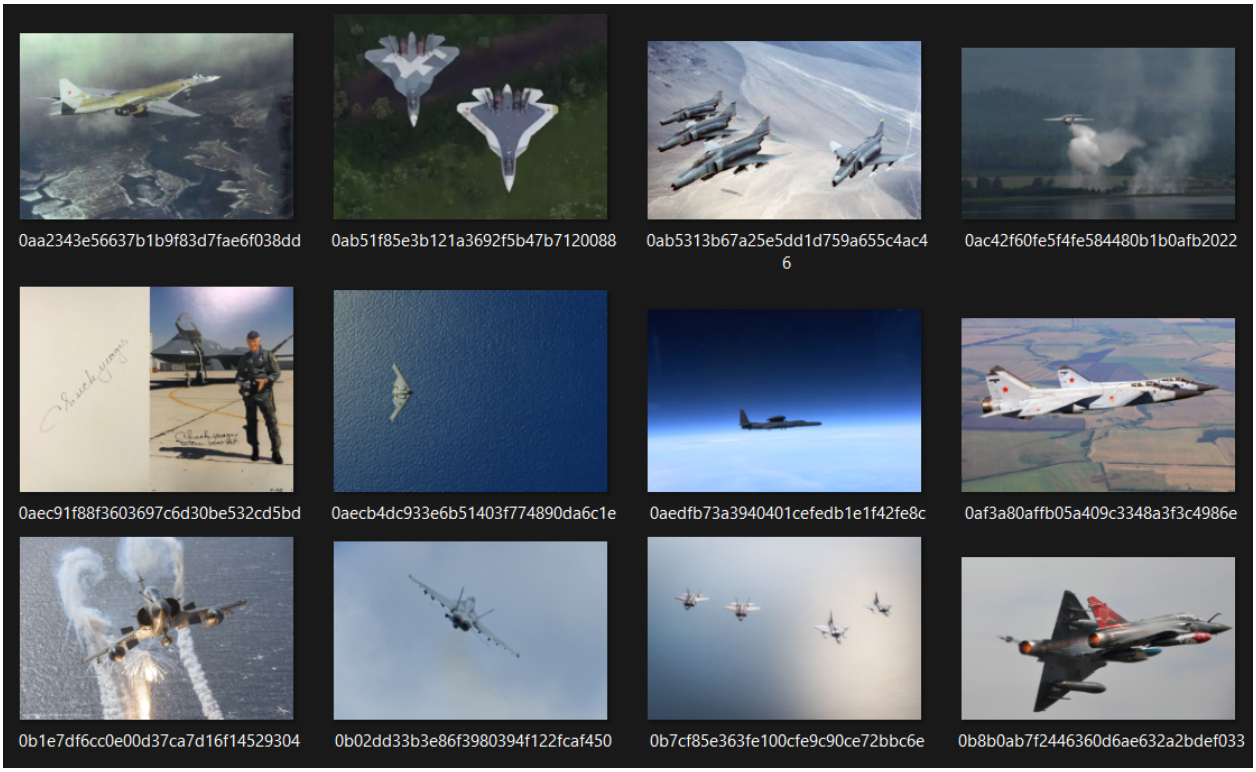

Figure 1. Sample images of the military aircrafts

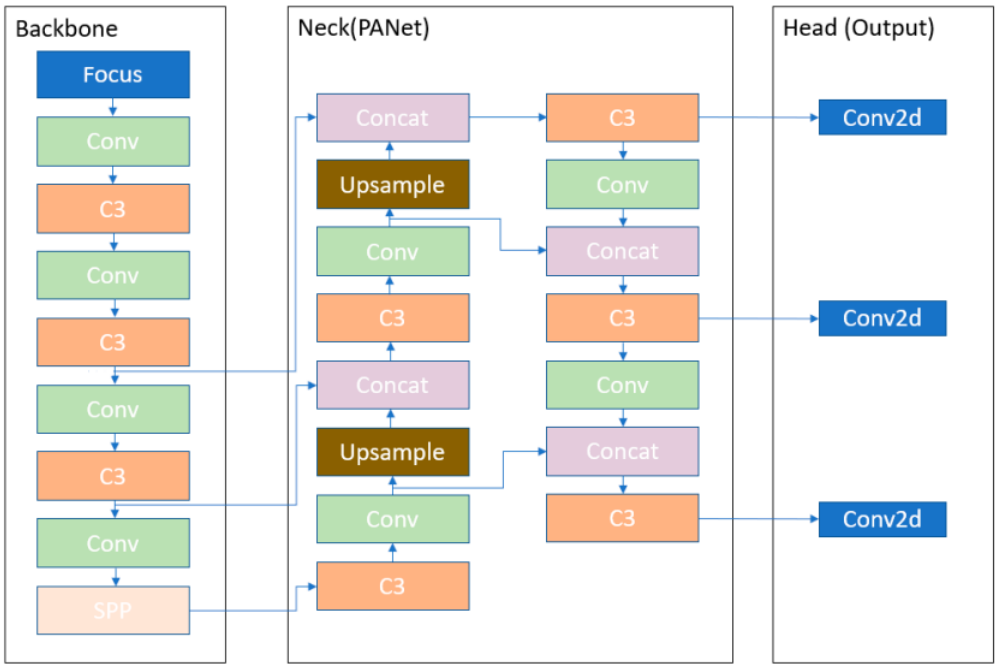

Figure 2. YOLOv5 Architecture [5]

the images (which we call as the smaller dataset here) for the training and validation of the system. The exact number of instances used for training and validation as well as the number of epochs the model was trained is reported in Table 1.

Sample training and validation batch of images from the

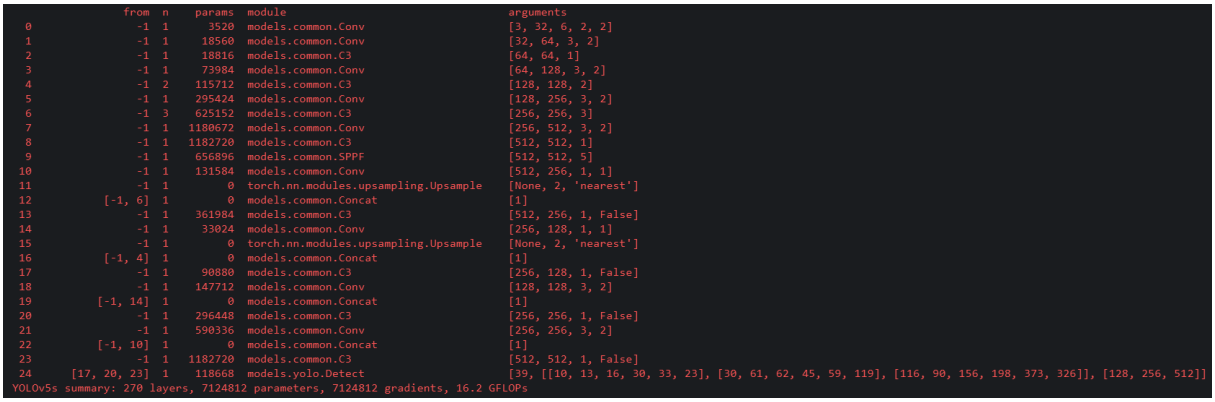

Figure 3. Layers of the implemented YOLOv5

Dataset	No. of Epochs	Training instances	Validation instances
Whole dataset	3	4020	1723
Small dataset	50	350	150

Table 1. Parameters for the performed experiments

whole dataset and the smaller dataset are represented by Figures 4, 5 and 6 and 7 respectively. The following Results section provides an insight on the results obtained for the described investigations.

To mention some of the failed attempts for some experiments, we also tried modifying the convolutional layers at the beginning of the network i.e backbone of the model or removing some layers from the neck of the model, but that did not work since the implementation is a close knit one.

4. Results

YOLOv5 comes with various out-of-the-box performance metrics which are automatically generated after every successful training run. These performance metrics consist of different types of losses and metrics and its explanation and how they are calculated. Here, the losses are calculated for both the training dataset and validation dataset respectively.

- **Box Loss -** The loss caused due to bounding box not exactly covering the object. It is also called bounding box regression loss and is of a mean squared error form. It is calculated for both training and validation instances given validation dataset is also provided. [6, 7]
- **Objectness Loss -** It is the loss that verifies the confidence of object i.e notifies about the wrong box object IOU predictions. IOU here means the overlap between two boundaries. It is used to evaluate how much does the predicted boundary of the provided sample overlap with the ground truth. This loss is calculated using the Binary Cross Entropy Loss formula.[6, 7]
- **Classification Loss -** It is the loss that occurs due to the wrong predictions of classes for the objects present in the image. This is calcualted in the form of Cross Entropyg Loss.[6, 7]
- **Precision -** This metrics means how precise and accurate the predictions are with respect to the ground truth i.e how correct the predictions are. The formula for precision is shown in Figure 8.[6, 7]
- **Recall: -** This metrics determines how reliable the positive predictions are i.e. how many of the true bounding box were predicted correctly. The formula for recall calculation can be seen in Figure 8.[6, 7]

mAP is the mean average precision score that compares the detected bounding box with that of the ground truth. The higher the score, the more accurate are the predictions. Here the average precision of all pictures of individual categories are calculated which is later averaged with respect to all the classes[6, 7]

- **mAP_0.5: -** It demonstrates the mAP at IoU (Intersection over Union) threshold of 0.5. [6, 7]
- **mAP_0.5:95 -** It represents the mAP when the threshold of IoU varies from the range of 0.5 to 0.95 in steps of 0.05.[6, 7]

For the whole dataset experiment while interpreting the results represented by Figure 10, it can be clearly seen that both the training and validation losses have a decreasing trend. Moreover, there is an increasing trend for precision, recall, mAP 0.5 as well as mAP 0.5:0.95. But it seems like

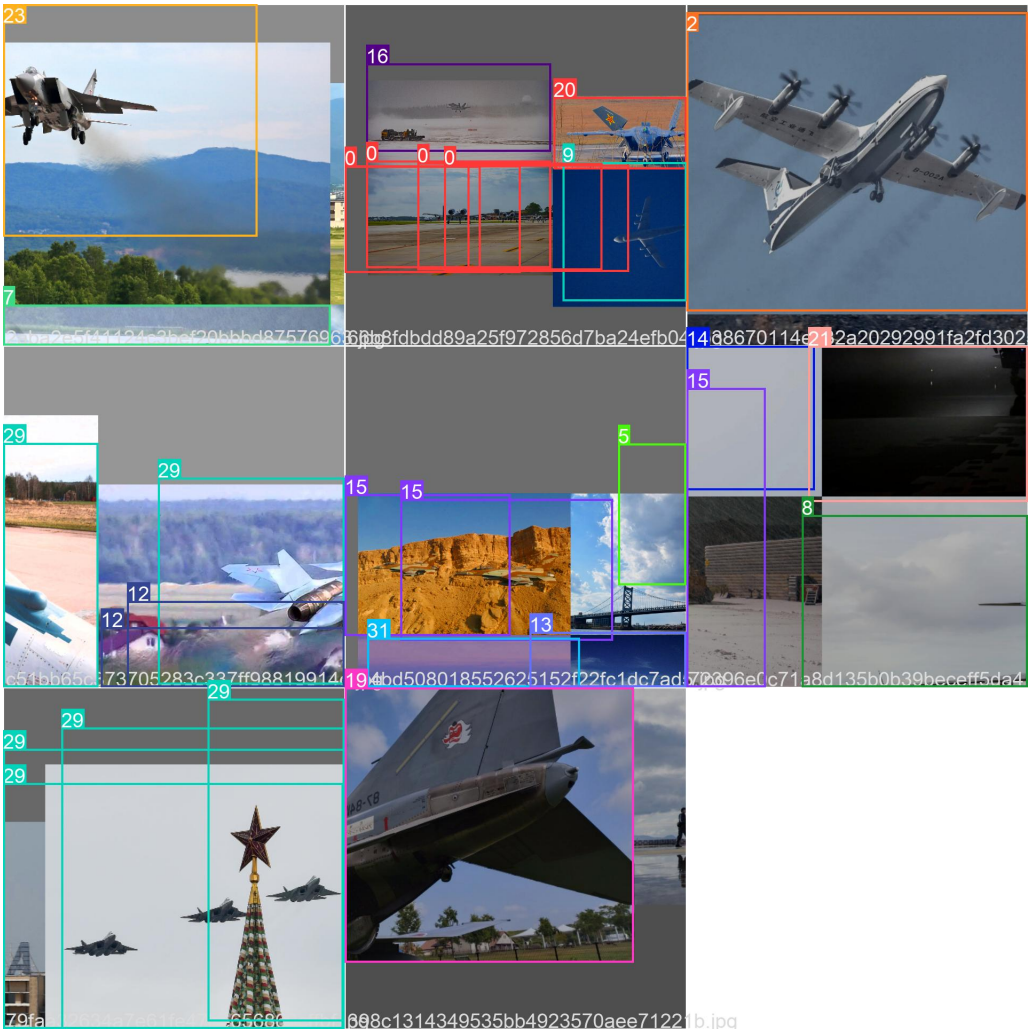

Figure 4. Batch of training images for the whole dataset

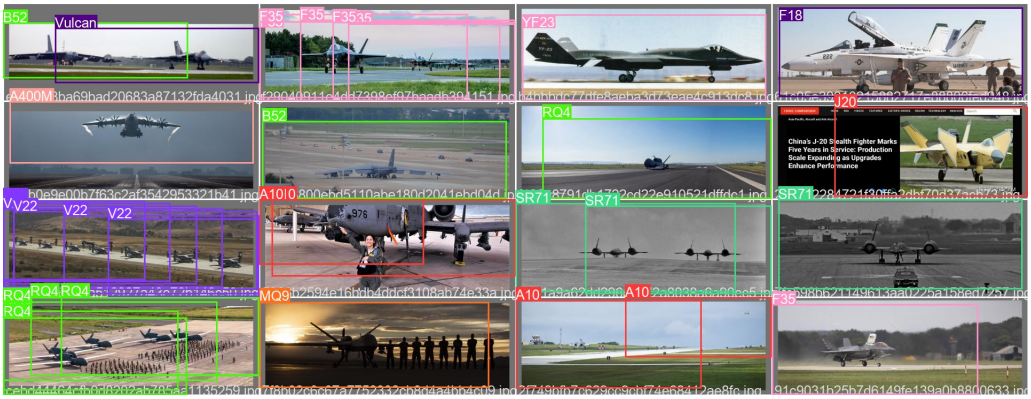

Figure 5. Batch of validation images for the whole dataset

the number of observations are quite less i.e. all the metrics could have further improved if we trained this data for greater number of epochs.

For the small dataset it is quite visible in Figure 11 that after a specific number of epochs the results for the precision, recall and mAP plumetted significantly. The previous interpretation also stands true for objectness and classification losses which deteriorate after a certain number of

Figure 6. Batch of training images for the small dataset

epochs. Only the box loss seems to have a decreasing trend. Thus, it can be safely said that this execution should have been stopped at the number of epochs where best results were obtained for this specific set of images provided for the training.

Thus, from the performed experiments, the results for small dataset trained for more number of epochs look more promising. Additionally, a better performance metrics can be obtained by training the model with the whole dataset for more number of epochs.

5. Future Work

There is always a chance of improving your work. Hence, following are some pointers on how the current implementation could be improved or experimented with in order to improve the performance of the detection and classification system.

- Balance the dataset in terms of instances for each class and then train it.
- Train the model for more number of epochs with more instances for each class.
- Perform more experiments by successfully modifying the dimensions of the existing layers or by removing them.
- Train the data using multiple versions of YOLOv5.
- Use a deconvolutional layer as the layer before the cross convolution layer in YOLOv5 or as the last layer of the model.
- Use a pixel shuffle layer as the layer before the cross convolution layer in YOLOv5 or as the last layer of the model.

Figure 7. Batch of validation images for the small dataset

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

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

TP = True positive

TN = True negative

FP = False positive

FN = False negative

Figure 8. Formula for precision and recall

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k$$

$$AP_k = \text{the AP of class } k$$

$$n = \text{the number of classes}$$

Figure 9. Formula for mean average precision

- Experiment with various loss functions like Logit, Sigmoid Focal Cross Entropy or Weighted Loss.

6. Conclusion

In this study, we used a variety of military aircraft images and their respective labels and performed their detection and classification. To achieve this, we converted the format of labels and sliced a smaller dataset from the whole dataset to perform some investigations. We passed this processed dataset and labels to the YOLOv5s deep learning model, performed two experiments and compared their respective results. From the results, it can be clearly seen that the results for the small dataset trained for more number of epochs were more promising. Hence, the model could be trained for the whole dataset for more number of epochs to further improve its performance and that can ultimately be used in the real-time systems for military aircraft categorization and detection in the battlefield.

7. Code

Following are some of the code citations that we referred for the implementation:

- YOLOv5 Github: `https://github.com/ultralytics/YOLOv5`
- Object Detection on Custom Dataset with YOLO (v5) using PyTorch and Python: `https://curiously.com/posts/object-detection-on-custom-dataset-with-yolo-v5-using-pytorch-and-python/`

References

[1] Lu Chen, LZ, and Jinming Liu. (2020). Aircraft Recognition from Remote Sensing Images Based on Machine Vision. Journal of Information Processing Systems , 16 (4), 795–808. https://doi.org/10.3745/JIPS.02.0136

[2] Hai-Jun Rong, Ya-Xin Jia, Guang-She Zhao (2014). Aircraft recognition using modular extreme learning machine, Neurocomputing, Volume 128, Pages 166-174, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2012.12.064.

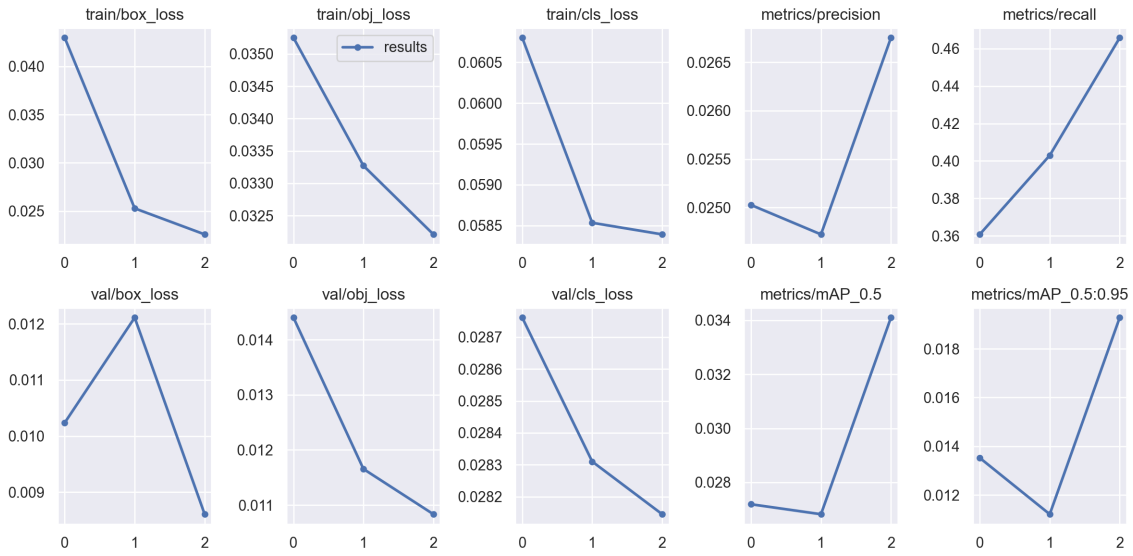

Figure 10. Performance metrics for the whole dataset trained for 3 epochs

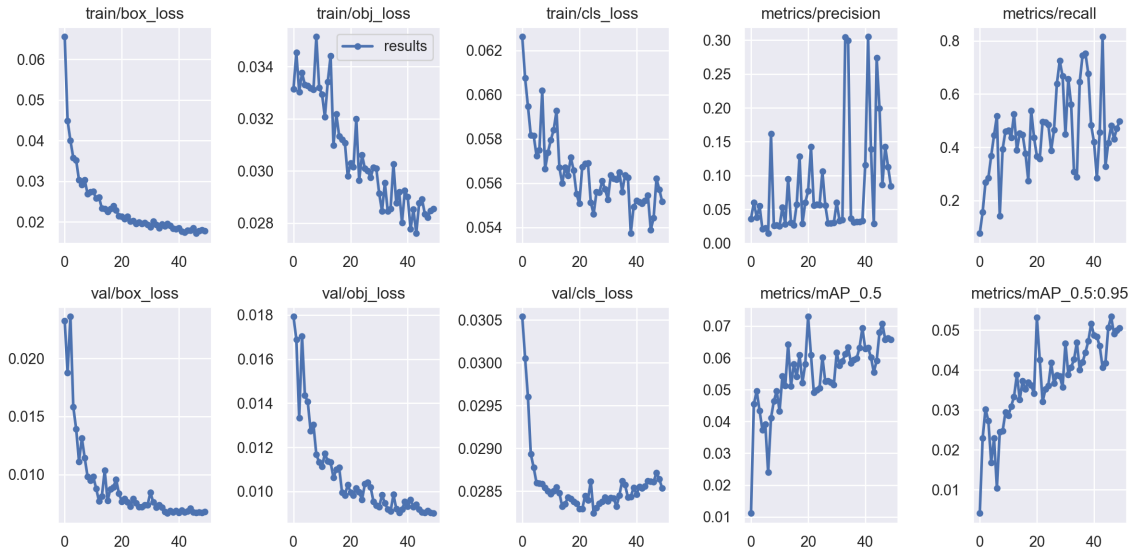

Figure 11. Performance metrics for small dataset trained for 50 epochs

[3] Roopa, K., and Ramamurthy, T.V. (2014). Aircraft Recognition System Using Image Analysis.

[4] S. F. Ali, J. Jaafar and A. S. Malik (2010). "Proposed technique for aircraft recognition in Intelligent Video Automatic Target Recognition System (IVA-TRs)," 2010 International Conference on Computer Applications and Industrial Electronics, pp. 174-179, doi: 10.1109/ICCAIE.2010.5735070.

[5] Nepal, U., and Eslamiat, H. (2022). Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. Sensors (Basel, Switzerland), 22(2), 464. https://doi.org/10.3390/s22020464

[6] Lihi Gur Arie (2022). The practical guide for Object Detection with YOLOv5 algorithm article in towards data science. https://towardsdatascience.com/the-practical-guide-for-object-detection-with-yolov5-algorithm-74c04aac4843

[7] Eléa Petton (2022. Object detection: train YOLOv5 on a custom dataset. https://blog.ovhcloud.com/object-detection-train-yolov5-on-a-custom-dataset/