Automated Identification of Military Aircraft from Images and Video

Dr. Raymond Scott Starsman
Defense Contract Audit Agency (DCAA)

raymond.s.starsman.civ@mail.mil

Bradford Lott
Air Force Operational Test and Evaluation Center (AFOTEC)

bradford.lott@us.af.mil

ABSTRACT

This study examines the performance of a single-pass detection algorithm to identify 41 different classes of military aircraft as well as identifies a data pipeline to include additional classes. The proposed Autonomous Aircraft Identification (AACID) method, capable of multi-object detection and near-real time predictions for video feed, achieves 82% test accuracy across the 41 classes which include military aircraft commonly used by the United States (US), Russian, Ukrainian, and Chinese Armed Forces as well as numerous defence partners of those nations. The US Department of Defense's ability to collect data exceeds its ability to analyse that data and convert it to actionable information. In addition to near-real-time predictions, we consider a scenario in which a Processing Exploitation and Dissemination (PED) analyst maintains a backlog of image and video files requiring analysis. AACID may assist the analyst in determining which files to review first by creating a "file-tag" including potential aircraft classes and quantities. This has the potential to improve intel product creation time. This work is a direct result of the U.S. Department of Defense Chief Digital and Artificial Intelligence Office's (CDAO) first-ever Create AI training program. All data used in this study is captured from publicly available sources.

Keywords: Multi-object detection, Aircraft identification, Processing Exploitation and Dissemination (PED)

1.0 INTRODUCTION

Data overload has been a consistent topic of concern for the United States Intelligence Community (IC) for over 20 years [1]. Passive data collection has led to an increased need for Processing, Exploitation, and Dissemination (PED) analysts to support the creation of Finished Intelligence Products (FINTEL). Major James Pineiro, United States Marine Corps (USMC) studied this data overload issue in 2020 and identified a common problem: "The proliferation of [autonomous reconnaissance systems] has expanded collections without a commensurate increase in PED. Unless steps are taken to automate information management, commanders risk information overload and decision paralysis." [2] In the same year Maj Pineiro identified this issue, the US Air Force (USAF) spent over $221M in research to support Project Maven: "…a rapid fielding Artificial Intelligence (AI) program to augment and automate PED for Full Motion Video (FMV) Tactical Unmanned Aerial Vehicles (TUAVs), Medium Altitude, High Altitude, and Wide Area Motion Imagery (WAMI) Intelligence, Surveillance and Reconnaissance (ISR) platforms..." [3] In the next two years, the US Air Force will provide an additional $90M in research funding for Project Maven although operational control will be assumed by the National Geospatial Intelligence Agency [3], [4], [5]. We contend that in order to leverage and maintain the AI tools we are developing, we need three resources: data, knowledge, and time. We need data to train and continuously re-train our predictive algorithms in a continuously changing battlespace. We need specialized knowledge both for the research analysts developing the AI tools as well as the intelligence analysts reviewing the predictions. We need time in each of these areas: to collect data, train our algorithms, and train our forces on how to build and utilize these tools. Maj Ricardo Colón USAF provides excellent insight into the knowledge component and why most if not all future PED analysts will require multi-source analysis capabilities. We refer the reader to Maj Colón's work for a comprehensive discussion of this knowledge component [4]. Our analysis focuses on the data and time components addressing the following two questions:

1) How do we ingest new data into our models?
2) How do we minimize time required to train and re-train our models?

Our study examines a scenario in which the AI's predictive goal is to identify military aircraft. We test a single-pass multi-object detection approach which accomplishes this task by providing near-real-time predictions for both live video as well as previously captured video requiring PED. Predictions are provided for both aircraft quantity and type. Our primary contributions are:

1) We identify and demonstrate a data pipeline in which we collect and process new images into our model training set.
2) We identify and demonstrate a model architecture capable of learning new aircraft classes without degrading existing predictive performance.
3) We identify the minimum amount of re-training required for our model to learn a new aircraft class, as well as demonstrate the trade-off between model training depth and predictive accuracy.

This work is a direct result of the U.S. Department of Defense Chief Digital and Artificial Intelligence Office's (CDAO) first-ever Create AI training program. All data used in this study are captured from publicly available sources. The remainder of this paper is organized as follows: Our Analytic Methods provides our current model architecture and training strategy; Our Results section demonstrates our model's capability to include additional aircraft classes and its performance for multiple levels of retraining; Our Conclusions section provides insights and considerations for PED analyst AI augmentation.

2.0 ANALYTICAL METHODS

The goal of this work was to both localize and classify military aircraft within an image. A number of approaches solve this challenge by addressing it in two stages: one stage locates an object in an image and then another stage classifies it. Common image localizers include: LeNet-5, Region-based Convolutional Neural Network (R-CNN), and Faster R-CNN [6], [7], [8]. Common image classification architectures include: AlexNet, VGGNet, InceptionNet, and ResNet [9], [10], [11], [12]. While these all represented a substantial advance in computer vision capability, classifying and locating an object in an image required multiple passes through the appropriate architectures.

A single-pass architecture called You Only Look Once (YOLO) was developed that could classify and locate multiple objects within an image in one pass [13]. This approach greatly increased the speed with which objects could be detected and located within an image and inspired considerable follow-on research and development. It was developed in a unique framework (Darknet) as were several new generations of the architecture.

The release of YOLO v5 [14] provides performance enhancements over the initial YOLO implementations as well as shifts the architecture to the much more common PyTorch framework. A high-level visualization of the YOLO v5 image processing architecture is shown in Figure 1.

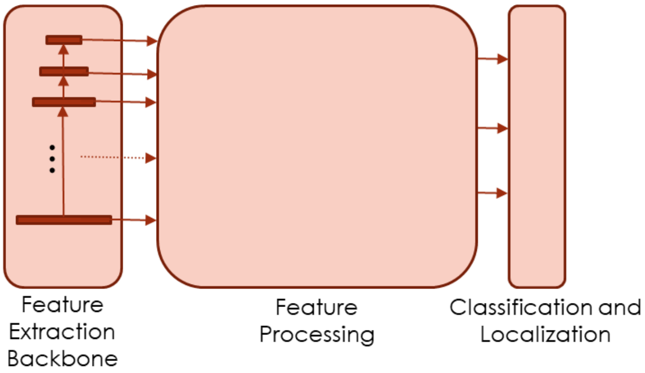

Figure 1: High-level YOLO v5 architecture.

The YOLO v5 architecture has three primary components: a feature extraction backbone, a feature processing layer, and a classification and localization layer. The feature extraction backbone is responsible for extracting object features from a provided image. This feature extraction occurs at multiple scales allowing the object detection to accommodate different object sizes within an image. The feature processing contains multiple Bi-directional Feature Pyramid Network (BiFPN) layers which processes the features from the backbone and performs multi-scale feature fusion. The classification and localization layer assembles the feature processing results and provides classification and localization predictions.

The base YOLO v5 system starts with a set of weights pre-trained on generic images and object classes. YOLO v5 provides a straightforward Application Programming Interface (API) to retrain the network on new image sets and classifications. The API provides a mechanism to freeze any number of the network's 24 meta-layers in order to leverage the initial training.

This effort was based on the work of a Kaggle user who trained YOLO v5 to classify the 40 different aircraft types in the baseline data set [15]. The size of the aircraft in the frame, whether the aircraft was flying or on the ground, and the orientation of the aircraft in the image all varied widely. Figure 2 shows 10 examples of the images. Notably, few of the images were from directly overhead as might be expected of military satellite imagery of aircraft on the ground. A typical result of classification and localization is shown in Figure 3.

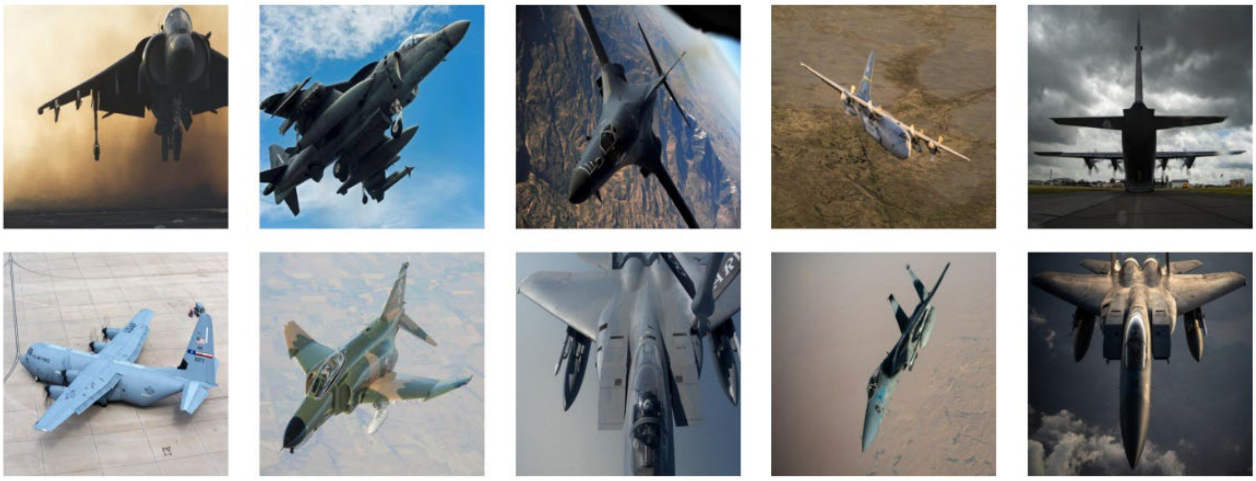

Figure 2: Image samples.

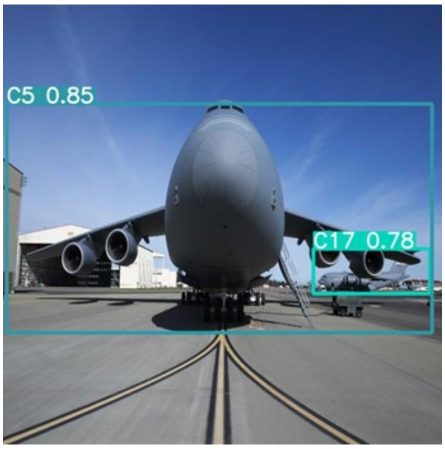

Figure 3: Sample classification and localization results.

As can be seen, the system draws a bounding box around the aircraft and classifies it along with a classification confidence level. This image demonstrates the ability of the algorithm to deal with multiple objects at different scales.

The data set originally had 40 aircraft classes with the number of examples of each varying from 106 images of the YF-23 to 867 images of the F-16. A plot of this is shown in Figure 4. Class imbalance can be a model performance issue with though, for the purpose of this project, that potential issue was not analyzed.

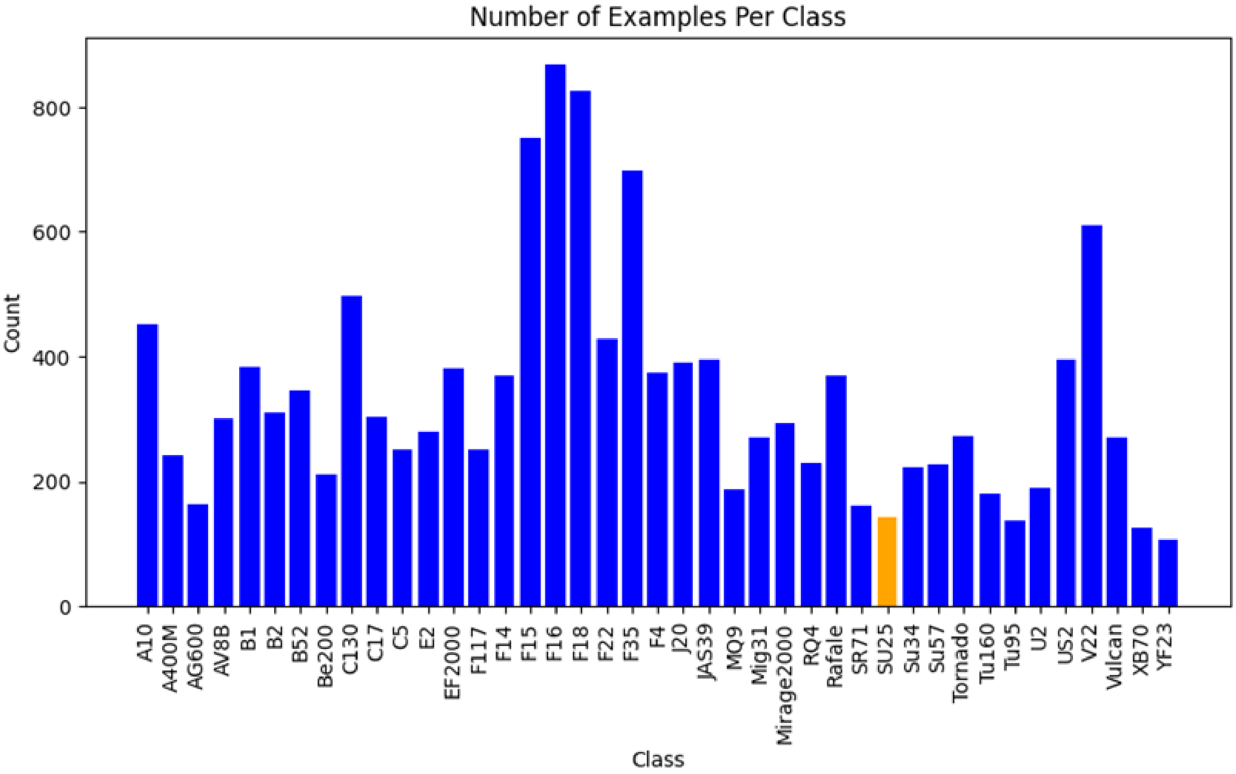

Figure 4: Comparison of number of class samples.

As one of the classes not existing within the data set was the SU-25 which is in broad use, particularly in the Russian invasion of Ukraine, our team felt that it would be useful to demonstrate how effectively a new class of aircraft could be added to the initial data set. Retraining a complex network can be quite time-consuming. One approach to reducing the time is to use a technique known as transfer learning whereby some number of layers in an existing network are frozen and not adjusted based on the new training data. Typically, this allows the low-level features that have been trained into the system (often at substantial processing effort) to be left unchanged and only the upper levels of the network would be trained thereby saving substantial training time.

Prior to retraining the network to add the SU-25, we ran the testing set of original images through the prediction to establish baseline classification performance. We used the F1 score as the accuracy metric and measured the accuracy across all 40 aircraft classes in the original data set. Overall, the classification accuracy was 0.77. Several classes yielded an accuracy of 100% (E2, SR-71, and XB-70). This is not surprising as these aircraft have very unique features which the network was able to learn. The lowest accuracies were on the MIG-31, YF-23, and SU-34 with respective accuracies of 0.29, 0.5, and 0.53.

The F1 score is a commonly used approach to measuring the performance of classifying systems. It is defined as the harmonic mean of the precision of a system (the number of correctly classified samples divided by the total number of samples placed in this class) and the recall of the system (the number of correctly classified samples divided by the actual number of samples in this class). It is often used to measure classification models as it balances precision (which can be skewed by a large number of false negatives) and recall which can be skewed by a large number of false positives.

We identified 140 images of the SU-25 from open sources on the internet and added them to the original data set. Of those, 115 were used for training and 25 were used for testing. Roboflow, an open source image processing tool, was used to label the new images. In addition to allowing the user to identify the bounding box and aircraft type within images, Roboflow completes the required formatting and preprocessing stages required to utilize the new instances in the YOLO model architecture.

3.0 RESULTS

YOLO v5 provides the capability of freezing any one of the 24 meta-layers in the model during training. Allowing a portion of the model to be retrained while retaining much of the original training work in the form of the lower-level model weights. This saves substantial training effort based on the number of frozen layers.

We began our effort by freezing all but the final (classification) layer. We trained the original network for 30 epochs and found that the ability to recognize the SU-25 was well below the average of the original accuracy metric (0.77). It was decided to freeze only the bottom 16 meta-layers of the original network and retrain with the addition of the SU-25 images for 30 epochs. As a result of this training, the test set of SU-25s was classified with an accuracy of 0.79 which was higher than the average for the other classes. Additionally, the overall classification accuracy was 0.78, slight improved over the baseline result. This demonstrated that we were able to add a class of aircraft and maintain the overall system accuracy.

The results of the training across the 41 classes are shown in Figure 5. The blue bar indicates the baseline model performance, the green bar indicates the new model performance, and the red bar shows the change between the two. They are arranged in improving performance where the aircraft class with the greatest classification improvement is to the left and class with degraded performance to the right.

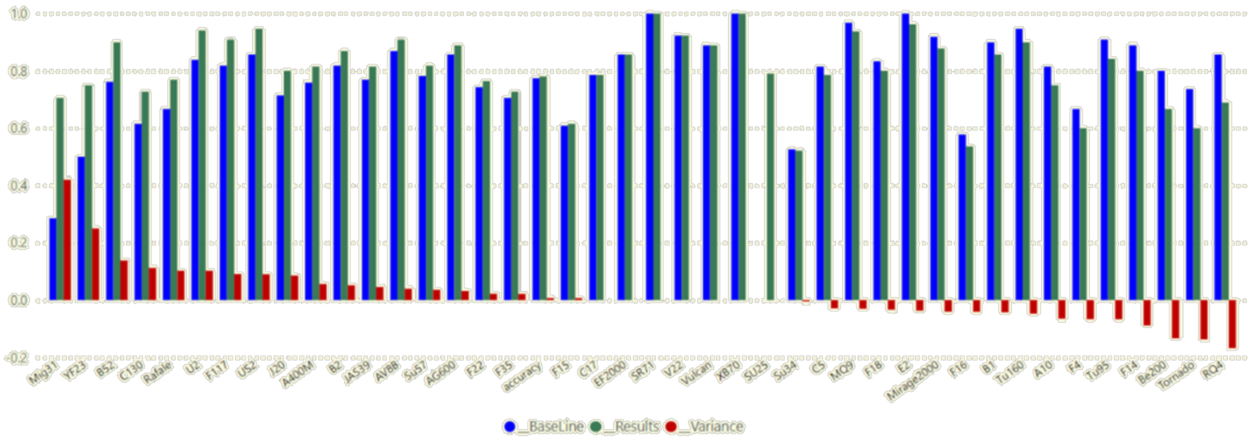

Figure 5: Comparison between baseline and SU-25 models.

The MIG-31 and YF-23 both were greatly improved whereas the RQ-4 and Tornado had some degradation. As the overall model accuracy was better than the baseline and the classification accuracy of the SU-25 was similar to an average aircraft classification, this model was deemed successful in adding the SU-25 class.

A final experiment was run to attempt to determine a proper layer freeze depth for a problem of this sort. As noted above, we already examined the impact of freezing 23 and 16 layers and found that 23 performed poorly whereas 16 performed satisfactorily. However, the freezing of 16 layers was simply an arbitrary choice that happened to provide satisfactory results.

We began a series of experiments to see if we could determine the optimal number of layers to freeze and still provide acceptable performance. We froze 2, 8, 16, 18, 23, and 24 layers in an attempt to develop a performance curve. Optimally, we would run an experiment on all possibilities of frozen layers, but there was insufficient time and computing resources available to do that. We retrained the network with the additional SU-25 images for 30 epochs and tracked the overall accuracy and of the accuracy for the new class added. Figure 6 shows the results of the training.

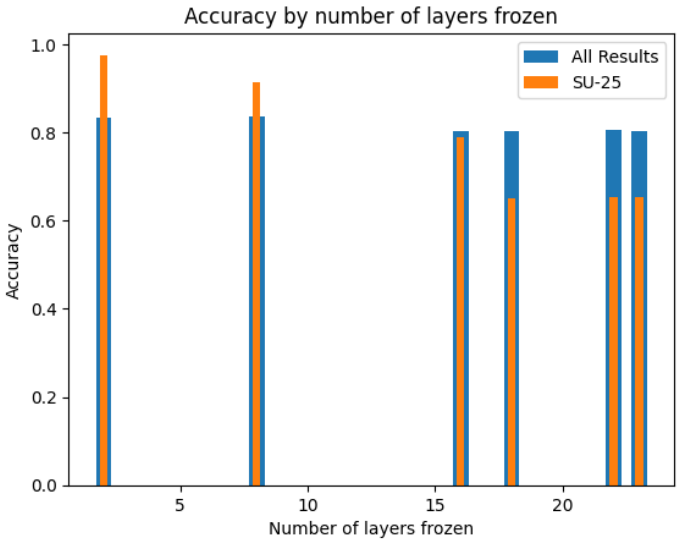

Figure 6: Classification accuracy versus number of frozen layers.

The wide, blue bar indicates the overall classification performance for non-SU-25 aircraft classes and the narrow orange bar indicates the performance on the added SU-25 class. We observe diminishing returns on accuracy with respect to the number of unfrozen layers. In the case of the SU-25, 18, 23, and 24 frozen layers yielded an unsatisfactory result of about 0.65. With 16 frozen layers, the performance of classifying the SU-25 achieved a satisfactory 0.79. Classification performance continues to improve as more layers are unfrozen and achieved an accuracy of 0.91 with 8 frozen layers and 0.98 with 2 frozen layers. This is likely due to the unique features present on the SU-25 and those augmenting the feature extractors.

4.0 CONCLUSION

The YOLO multi-object detection model may effectively support intelligence analysts suffering from data overload by identifying objects of interest and quantities within pictures and videos requiring review. Publicly available resources exist both to incorporate new data into this model and to train this model. If the baseline model accuracy is acceptable for a given use case, technical practitioners may consider retraining approximately one third of the original model architecture when performing a similar task to the original baseline model. If increased accuracy is desired, or performing a dissimilar task compared to the original model, technical practitioners may consider retraining approximately two thirds of the original model architecture. Future applications may explore this model applied to other objects of interest. Future research may examine, in higher resolution, the relationship between accuracy and the amount of retraining performed; computational complexity of this specific algorithm as a function of the number of frozen layers; and whether the respective training validation and test accuracies have any correlation to their respective image distributions on the National Imagery Interpretability Rating Scale.

5.0 ACKNOWLEDGEMENTS

R.Scott Starsman – I would like to thank my command, the Defense Contract Audit Agency (DCAA) for providing the time to participate in this work, the CDAO for providing the CreateAI course, and my wife Stacey for putting up with seeing only the back of my head while completing this effort.

6.0 REFERENCES

[1] MacDonald, Oettinger "Information Overload: Managing Intelligence Technologies" Harvard International Review. 24.3 (2002) 44-48.

[2] Pineiro, James D. "Gaining a Cognitive Advantage: Artificial Intelligence as a Decision Support System" Defense Technical Information Center (2020).

[3] The United States Department of Defense "Fiscal Year 2020 Budget" Defense-Wide Justification Book.

[4] Colón, Ricardo D. "Artificial Intelligence and PED: Preparing Humans for Human-Machine Teaming" Defense Technical Information Center (2018).

[5] Strout, Nathan "Intelligence agency takes over Project Maven, the Pentagon's signature AI scheme" C4ISRNET (2022).

[6] LeCun, Yann, et al. "Gradient-Based Learning Applied to Document Recognition" Proceedings of the IEEE (1998): 1-46.

[7] Girshick, Ross, et al. “Region-based convolutional neural networks for object detection.” IEEE Transactions on Pattern Analysis and Machine Intelligence 38.1 (2016): 144-154.

[8] Ren, Shaoqing, et al. “Faster R-CNN: Towards real-time object detection with region proposal networks.” IEEE Transactions on Pattern Analysis and Machine Intelligence 39.6 (2017): 1137-1149.

[9] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[10] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 [CS.cv].

[11] Szegedy, C., Liu, W., Jia, Y., et al (2014). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1448-1456).

[12] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. arXiv:1512.03385 [CS.cv].

[13] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi (2016), You Only Look Once: Unified, Real-Time Object Detection, arXiv:1506.02640v5 [cs.CV].

[14] Jocher, Glenn et al. ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations, Apr. 2021.

[15] Blackadder97, https://www.kaggle.com/code/blackadder97/militaryaircraftdetection-with-yolov5