# Detecting Adversary Military Assets on Autonomous Edge Devices using Deep Learning

by Sean Brendan Smith

B.A. in International Politics, May 2016, The Catholic University of America
M.S. in Cybersecurity - Management of Secure Information Systems, May 2019, George Mason University
M.P.S. in Technology Management, May 2022, Georgetown University

A Praxis submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Engineering

January 9, 2026

Praxis directed by

John H. Kuk
Professorial Lecturer in Engineering and Applied Science

The School of Engineering and Applied Science of The George Washington University certifies that Sean Brendan Smith has passed the Final Examination for the degree of Doctor of Engineering as of November 18, 2025. This is the final and approved form of the Praxis.

**Detecting Adversary Military Assets on Autonomous Edge Devices using Deep Learning**

Sean Brendan Smith

Praxis Research Committee:

John H. Kuk, Professorial Lecturer in Engineering and Applied Science, Praxis Director

Amir Etemadi, Adjunct Professor of Engineering and Applied Science, Committee Chair

Mohamed Belkhayat, Professorial Lecturer in Engineering and Applied Science, Committee Member

© Copyright 2025 by Sean Brendan Smith
All rights reserved

## Dedication

This praxis is dedicated to my family. Your sacrifices, encouragement, prayers, and unwavering support made this achievement possible.

## Acknowledgements

Thanks to my research advisor, Dr. John H. Kuk. Your dedication, unwavering support, insight, and friendship throughout this journey have been instrumental in the development of this research.

Thanks to the faculty of The George Washington University School of Engineering and Applied Sciences and my peers in cohort DA2.

## Abstract of Praxis

### Detecting Adversary Military Assets on Autonomous Edge Devices using Deep Learning

The research in this praxis explores the potential use of deep learning AI on small, unmanned aircraft systems (sUAS) to detect adversary military assets in electronic warfare (EW) disrupted environments. This praxis identifies several impacts of EW to the command and control of sUAS in military applications. Several research questions used in this praxis investigate various constraints that hardware and environmental conditions have on AI deployed on edge devices. This praxis implements a convolutional neural network (CNN) using a medium size version of the 8$^{th}$ generation of the You Only Look Once model architecture. This model is trained on images of modern Russian military assets including main battle tanks and infantry fighting vehicles. This praxis also uses a dedicated test set of images as well as simulated images generated in an Unreal Engine test environment. The experiments in this praxis are conducted on an NVIDIA Jetson Orin Nano Super Developer Kit to simulate the limitations found on edge devices. The results of this praxis demonstrate the potential application of a lightweight CNN to detect adversary military vehicles from an edge device to mitigate the disruption of EW on sUAS. This research contributes to the future development of artificial intelligence equipped edge devices in the defense sector, particularly with efforts like the U.S. Department of Defense (DOD) Replicator Initiative. Future research may focus on robust dataset generation, model refinement, integration with autonomous flight control systems, and integration with military battlefield intelligence systems.

## Table of Contents

Dedication ..... iv

Acknowledgements ..... v

Abstract of Praxis ..... vi

List of Figures ..... xii

List of Tables ..... xiii

List of Equations ..... xiv

List of Acronyms ..... xv

Glossary of Terms ..... xvii

Chapter 1—Introduction ..... 1

1.1 Background ..... 1

1.2 Research Motivation ..... 2

1.3 Problem Statement ..... 3

1.4 Thesis Statement ..... 4

1.5 Research Objectives ..... 5

1.6 Research Questions and Hypotheses ..... 6

1.6.1 Research Questions ..... 6

1.6.2 Hypotheses ..... 7

1.7 Scope of Research ..... 7

1.8 Research Limitations ..... 8

1.9 Organization of Praxis ..... 8

Chapter 2—Literature Review ..... 9

2.1 Introduction ..... 9
2.2 Electronic Warfare (EW) and Small Unmanned Aircraft Systems (sUAS) ... 10
2.2.1 The Role of EW and sUAS in Modern Military Applications ..... 10
2.2.2 Miniaturization and Capability Advancements of sUAS ..... 11
2.2.3 Affordability and Tactical Impact in Contemporary Conflicts ..... 12
2.2.4 Policy and Strategic Framework for sUAS ..... 13
2.2.5 Electronic Warfare and Counter-Drone Defenses ..... 14
2.2.6 Electronic Warfare in the Russia-Ukraine Conflict ..... 14
2.2.7 Asymmetric Advantages of AI Augmented Drones ..... 15
2.2.8 Security Threats Facing Drones ..... 16
2.3 AI/ML and sUAS ..... 16
2.3.1 Convolutional Neural Networks on Edge Devices ..... 16
2.3.2 Artificial Intelligence Based Control for Drones ..... 17
2.3.3 YOLOv4-Tiny for Real-Time Detection on Edge Devices ..... 18
2.3.4 CNN Object Detection Features and Optimizations on UAVs ..... 19
2.3.5 Evaluation of Deep Learning Models on UAVs ..... 19
2.3.6 Small Scale Object Detection for UAVs ..... 20
2.3.7 Resolution and Power Consumption Trade-Offs for Image Detection .. 20
2.3.8 Real-Time Object Detection of Vehicles on Edge Devices with AI ..... 21
2.4 You Only Look Once (YOLO) ..... 21
2.4.1 YOLO Model Architectures ..... 22
2.4.2 Robust Real-Time Object Detection with YOLO Architecture ..... 22
2.4.3 YOLO for Unmanned Aerial Vehicle Deployment ..... 23

2.4.4 YOLOv8 Selection for Edge Applications ..... 25
2.5 Training, Optimization, Simulation ..... 29
2.5.1 Optimization of YOLOv8 to Minimize Computational Overhead ..... 29
2.5.2 Small Object Detection with YOLO ..... 29
2.5.3 Data Generation with AirSim and Unreal Engine ..... 30
2.5.4 Simulated Test Environments using Game Engines ..... 31
2.6 Analysis ..... 32
Chapter 3—Methodology ..... 34
3.1 Introduction ..... 34
3.2 Research Design and Approach ..... 34
3.3 Prototyping ..... 35
3.4 Data Sources ..... 36
3.4.1 Data Cleaning and Structure ..... 37
3.4.2 Data Preparation ..... 38
3.5 Model Research and Selection ..... 39
3.5.1 YOLO Variants for Object Detection ..... 39
3.5.2 YOLOv8 Architecture ..... 40
3.6 Experiment Environment ..... 41
3.6.1 Desktop Coding Environment ..... 41
3.6.2 Nvidia Jetson Orin Nano Test Environment ..... 41
3.7 Simulated Environment ..... 43
3.8 Equations ..... 45
3.8.1 Bounding Box ..... 45

3.8.2 Confusion Matrix ..... 45
3.8.3 Precision ..... 46
3.8.4 Recall ..... 46
3.8.5 F1-Score ..... 47
3.8.6 Intersection over Union (IoU) ..... 47
3.8.7 Mean Average Precision ..... 47
Chapter 4—Results ..... 49
4.1 Introduction ..... 49
4.1.1 Limitations and Constraints ..... 49
4.2 Dataset ..... 50
4.3 Data Cleaning ..... 50
4.3.1 Optimization, Data Imbalance ..... 51
4.4 Model Training ..... 53
4.5 Deployment on Edge Computer ..... 57
4.6 Object Detection in Simulated Environment ..... 59
4.7 Hypothesis and Research Question Results ..... 61
4.7.1 Research Question 1 ..... 62
4.7.2 Hypothesis 1 ..... 66
4.7.3 Research Question 2 ..... 68
4.7.4 Hypothesis 2 ..... 69
4.7.5 Research Question 3 ..... 70
4.7.6 Hypothesis 3 ..... 71
Chapter 5—Discussion and Conclusions ..... 73

5.1 Discussion ..... 73

5.2 Conclusions ..... 74

5.3 Contributions to Body of Knowledge ..... 75

5.4 Recommendations for Future Research ..... 76

5.5 Ethical Considerations ..... 77

References ..... 79

Appendix A ..... 90

1 Oryx 2022 Russia Ukraine War Dataset (Kaggle) ..... 90

2 Military_Vehicles Dataset (Hugging Face) ..... 90

3 War_Tech_2.0 Detection Objects Dataset (Kaggle) ..... 90

Appendix B ..... 91

1 Link to AI/ML Code Repository (GitHub) ..... 91

## List of Figures

Figure 1: Vehicle Classes ..... 38

Figure 2: NVIDIA Jetson Orin Nano Super Developer Kit ..... 43

Figure 3: Squad SDK Running in Unreal Engine v4.27 ..... 44

Figure 4: Bounding Box Example ..... 45

Figure 5: Confusion Matrix Example ..... 46

Figure 6: Oryx 2022 Russia Ukraine War Dataset, Examples of Destroyed Vehicles ... 51

Figure 7: Oryx 2022 Russia Ukraine War Dataset, Example of Intact Vehicles ..... 51

Figure 8: YOLOv8x (extra-large) Training Results (25 epochs) ..... 56

Figure 9: YOLOv8m (medium) Training Results (100 epochs) ..... 56

Figure 10: YOLOv8m Test Split Predictions Sample ..... 58

Figure 11: YOLOv8m Test Split Results, Confusion Matrix ..... 59

Figure 12: Unreal Engine v4.27, Lighting Conditions Tests ..... 60

Figure 13: YOLOv8m Model Confusion Matrix for Normal Lighting Test ..... 63

Figure 14: YOLOv8m Model Confusion Matrix for Low-Lighting Test ..... 66

Figure 15: NVIDIA Jetson Orin Nano Super Developer Kit Power Modes ..... 68

## List of Tables

Table 1: Dataset Pre-Class Balance ..... 52

Table 2: Dataset Post-Class Balance ..... 53

Table 3: YOLOv8x and YOLOv8m Training Hyperparameters ..... 55

Table 4: YOLOv8m Model Results ..... 57

Table 5: YOLOv8m Individual Class Results ..... 58

Table 6: Research Question 1, Normal Light Conditions ..... 63

Table 7: Research Question 1, Low-Light Conditions ..... 65

Table 8: Research Question 2 Results ..... 69

Table 9: Research Question 3 Results ..... 71

## List of Equations

Equation 1: Bounding Box ..... 45

Equation 2: Precision ..... 46

Equation 3: Recall ..... 46

Equation 4: F1 Score ..... 47

Equation 5: Intersection Over Union (IoU) ..... 47

Equation 6: Mean Average Precision (mAP) ..... 47

Equation 7: Mean Average Precision 50 (mAP50) ..... 48

Equation 8: Mean Average Precision 50-95 (mAP50-95) ..... 48

## List of Acronyms

| | |
|---|---|
| ADA2 | All-Domain, Attritable-Autonomous |
| AFU | Armed Forces of Ukraine |
| APC | Armored Personnel Carrier |
| AI | Artificial Intelligence |
| C2 | Command-and-Control |
| CNN | Convolutional Neural Networks |
| C-sUAS | Counter Small Unmanned Aircraft System |
| DDIL | Denied, Degraded, Intermittent, or Limited (Bandwidth) |
| DHS | Department of Homeland Security |
| DOD | Department of Defense |
| DSAIC | Defense Systems Information Analysis Center |
| EW | Electronic Warfare |
| IFV | Infantry Fighting Vehicle |
| IoU | Intersection Over Union |
| ISR | Intelligence, Reconnaissance, and Surveillance |
| LAV | Light Armored Vehicle |
| LAWS | Lethal Autonomous Weapon Systems |
| NGO | Non-government Organization |
| mAP | Mean Average Precision |
| mAP50 | Mean Average Precision 50 |
| mAP50-95 | Mean Average Precision 50-95 |

| | |
|---|---|
| MBT | Main Battle Tank |
| ML | Machine Learning |
| RUSI | Royal United Services Institute |
| sUAS | Small Unmanned Aircraft System |
| UAV | Unmanned Aerial Vehicle |
| UE | Unreal Engine |
| U.S. | United States |

## Glossary of Terms

**Attritable:** Term used by DOD to describe specific sUAS capabilities found in the Replicator Initiative that include low cost to manufacture and design for use in contested environments where risk of higher attrition of sUAS platforms may be acceptable in comparison to traditional and more expensive UAV platforms (Milonas et al., 2024).

**Convolutional Neural Networks (CNN):** A form of feedforward deep learning model comprised of layers to adaptively learn features from data. This architecture is used as a backbone in computer vision processing (De Oliveira Barbosa et al., 2025).

**Denied, Degraded, Intermittent, or Limited (DDIL) Environment:** A category of military communication environment with inconsistent communications and data connectivity (Busch, 2018).

**Deep Learning:** A form of neural network model that leverages labeled data and multiple convolutional layers for machine learning to identify significant features for classification (Ma et al., 2023).

**DOD Replicator Initiative:** An initiative announced by the U.S. DOD Defense Innovation Unit to solicit the research, development, and deployment of highly scalable and attritable autonomous systems such as sUAS devices equipped with artificial intelligence capabilities (U.S. Department of Defense, 2024).

**Electronic Warfare (EW):** A military technology countermeasure leveraging the electromagnetic spectrum to intercept, interfere, jam, or spoof electronic communications signals (Molloy, 2024).

**Intelligence, Reconnaissance, and Surveillance (ISR):** A category of military capabilities pertaining to technologies that provide methods to collect visual or signal data for analysis. Commonly used in reference to onboard collection capabilities of drones (Busch, 2018).

**Lethal Autonomous Weapon System (LAWS):** U.S. Government category for describing weapon system technologies that provide lethal capabilities with limited direct control by a human (Sayler, 2024).

**NVIDIA Jetson Orin Nano:** A compact, lightweight computer designed by NVIDIA for deployment of AI on edge devices (NVIDIA, 2025).

**One Stage Detector:** Form of computer vision model architecture that processes an image through a single convolutional neural network (CNN) generally providing faster inference speed (Hakani & Rawat, 2024).

**Small Unmanned Aircraft System(sUAS):** A category of unmanned aircraft typically under 55 pounds with various uses across infrastructure, public safety, border security, and defense (U.S. Department of Homeland Security, 2019).

**Two Stage Detector:** Form of computer vision model architecture that uses two stages to first generate proposals of object locations in an image before conducting classification of the proposed locations with a separate neural network, generally resulting in higher accuracy but lower inference speed (Hakani & Rawat, 2024).

**Unreal Engine (UE):** Three-dimensional computer modeling software by Epic Games for the development of virtual environments. Commonly used in media creation, video games, and simulated environments (Krüger et al., 2024).

**Unmanned Aerial Vehicle (UAV):** A category of aircraft that does not carry a human operator onboard and is remotely piloted (Milonas et al., 2024).

**You Only Look Once (YOLO):** A deep learning AI algorithm that was introduced by Redmon et al. (2016) that advanced real-time object detection. The YOLO model uses a one stage detector to provide fast inference speeds while maintaining high levels of precision (Redmon et al., 2016).

# Chapter 1—Introduction

## 1.1 Background

The proliferation of small, unmanned aircraft systems (sUAS) has had a significant impact on modern military capability and strategy (Gerstein & Leidy, 2024; Pledger, 2021). The low cost and high availability of sUAS have increased the use of these devices in modern military conflicts by nation states and non-nation states (Pledger, 2021). This unmanned technology has been used to provide different tactical capabilities including intelligence, surveillance, and reconnaissance (ISR) in addition to kinetic strike capabilities when combined with munitions (Pledger, 2021). Advancements to commercially available drones over the past decade have resulted in lower costs and increased accessibility to this technology (U.S. Department of Defense, 2024). More importantly, from an operational or tactical perspective, these advancements have increased capabilities supporting various military and paramilitary applications (U.S. Department of Defense, 2024). The 2022 invasion of Ukraine demonstrated the increased use of sUAS paired with munitions to remotely target adversary military assets (Molloy, 2024). As documented by Finer and Shimer (2025) and Molloy (2024), the Armed Forces of Ukraine (AFU) have manufactured hundreds of thousands of inexpensive sUAS to repel substantially more expensive adversary Russian military assets like tanks and other armored vehicles. The implementation of sUAS in modern military applications has reshaped military strategy and has demonstrated several key advantages such as cost-effectiveness and asymmetric tactical capabilities (Busch, 2018; Molloy, 2024).

Electronic Warfare (EW) has been used as a highly effective countermeasure against sUAS devices (Watling & Reynolds, 2023). EW countermeasures provide the capability to interfere with or completely sever signal communications between the sUAS device and the operator and demonstrates one of the most significant issues to the deployment of sUAS devices in military applications (Kallenborn & Plichta, 2024). According to Kallenborn and Plichta (2024), most counter-drone defenses are comprised of EW capabilities. However, advancements in autonomous drone technologies may potentially reduce reliance on external command-and-control of drones in EW contested environments (Busch, 2018; Kallenborn & Plichta, 2024).

### 1.2 Research Motivation

The motivation of this praxis is heavily influenced by the requirements identified in the U.S. Department of Defense (DOD) Replicator Initiative as well as the DOD Data, Analytics, and Artificial Intelligence Strategy. These publications identify the strategic importance that the rapid deployment of AI augmented edge devices will have in contested environments. In particular, the DOD Replicator Initiative specifically identifies the strategic importance of deploying large quantities of attritable low-cost autonomous edge devices with AI capabilities necessary to counter adversary military assets (U.S. Department of Defense, 2023, 2024).

In addition to the requirements identified by U.S. DOD, several military capabilities and tactics implemented by the AFU have influenced the development of this praxis. In particular, the impacts of the 2022 Russian invasion of Ukraine highlight several military applications for sUAS devices as well as the significant impact that EW countermeasures have on disrupting signal communications between drones and their

operators in contested environments (Molloy, 2024). Insights from the military application of sUAS devices in Ukraine combined with the requirements identified in the DOD Replicator Initiative influence the research in this praxis to provide research and development toward the strategic implementation of AI on sUAS devices and mitigate the impact of EW in this domain.

### 1.3 Problem Statement

*Electronic warfare disrupts drone communication, causing 90% reduction in command-and-control effectiveness in contested environments and increasing operational risk and costs by an equivalent percentage (Finer and Shimer, 2025; Molloy, 2024; Watling & Reynolds,2023).*

The 2022 invasion of Ukraine highlighted that EW significantly hindered the use of sUAS devices in contested environments (Watling & Reynolds, 2023). The Russian use of EW countermeasures caused over a 90% reduction in command-and-control effectiveness of Ukrainian sUAS devices through signal interference resulting in loss of signal communications between drones and operators (Molloy, 2024; Watling & Reynolds, 2023). Many of the sUAS devices deployed in Ukraine relied on sustained communication between the device and the operator to transmit live video and receive command-and-control signals. Disruption to either of these signals resulted in reduced battlefield awareness or complete loss of control of the sUAS (Watling & Reynolds, 2023). As reported by Finer and Shimer (2025), an estimated loss of approximately 10,000 drones per month were attributed to deployment of EW countermeasures by Russian forces to disrupt the command-and-control capability of sUAS. Insufficient autonomous capabilities as well as heavy reliance on sustained communication with an

operator limit the effectiveness that sUAS devices may have in military applications in contested environments (O'Conner, 2023). This limitation highlighted a critical issue facing sUAS devices in military applications that could result in reduced battlefield awareness, delayed critical decision making, increased risk and cost, and the prevention of the completion of critical military missions (Gerstein & Leidy, 2024).

### 1.4 Thesis Statement

*A Convolutional Neural Network tailored for edge devices will enhance drone resilience to electronic warfare by over 90%, enabling detection and identification of adversarial military assets in electronic warfare-contested environments (Molloy, 2024; Sayler, 2024; Watling & Reynolds, 2023).*

The development of a predictive detection model in the form of Python code designed to function on edge computing hardware with limited memory and compute resources may significantly decrease the effectiveness of electronic warfare (EW) on small, unmanned aircraft systems (sUAS) in military applications (Lee et al., 2024). Moreover, the implementation of this model for use onboard sUAS may potentially support the requirements identified in the DOD Replicator Initiative to enhance attritable drones with AI capabilities for military applications in EW disrupted environments (U.S. Department of Defense, 2024; Milonas et al., 2024; O'Conner, 2023). Such a model may support potential military applications in Eastern Europe and East Asia (U.S. Department of Defense, 2024; O'Conner, 2023).

As part of this praxis, a lightweight convolutional neural network (CNN) is implemented to detect adversary military assets. This lightweight CNN is optimized to function on an edge device, making it potentially suitable for deployment on sUAS. The

purpose of this CNN is to mitigate the impact that EW has on the command-and-control of sUAS in military applications. The automated detection of adversary military assets like armored vehicles may be a critical component to augment sUAS and potentially enable pairing with automated flight controllers to traverse EW disrupted environments. Bypassing the need for manual visual detection of adversary assets by a human operator, the effect of EW disruption on sUAS devices may potentially be reduced by leveraging automated detection. By circumventing the impact of EW to the visual component of the command-and-control of an sUAS device, this model may potentially support a significant increase in sUAS resilience against EW contested environments. In addition to military application, the deliverable of this praxis may have potential application in non-defense industries like search and rescue and disaster response to equip resource constrained small drones the capability to detect and identify individuals or hazards in communication-limited environments.

### 1.5 Research Objectives

There are several research objectives in this Praxis. The first research objective is to identify the prevalence of sUAS in military applications and strategic planning. The second research objective is to identify the impact that EW has on the deployment of sUAS and the potential for artificial intelligence to augment the command-and-control of sUAS devices in military applications. The third research objective is to identify existing research and development of artificial intelligence models for use on sUAS devices. The fourth research objective is to identify an efficient CNN architecture suitable for deployment on an edge device limited by computational, memory, and power constraints. The fifth research objective is to identify model training optimization methods for a CNN

model that may be able to provide a significant increase in sUAS resiliency with consideration to memory and computational limitations without causing a significant reduction in model speed or detection accuracy. The sixth research objective is to identify methods to test and evaluate the real-time detection capabilities of a CNN model deployed on an edge device to include testing within simulated environments. The final research objective for this praxis is to identify areas for future contributions to the advancement of implementing AI on edge devices in military applications.

## 1.6 Research Questions and Hypotheses

There are several research questions and corresponding hypotheses that are addressed in this praxis. These research questions and hypotheses identify the relationship between computational limitations and the performance of CNN models for object detection on edge devices. The research questions are tested on an NVIDIA Jetson Orin Nano Super Developer Kit as well as tested with visual data generated from a simulated environment leveraging the Squad Software Development Kit (SDK) within Unreal Engine v4.27. The combination of these platforms provides a mechanism to simulate various environmental scenarios as well as computational limitations found on edge devices.

### 1.6.1 Research Questions

**RQ1:** What is the impact to real-time object detection accuracy in low light conditions using CNNs on edge devices?

**RQ2:** How do reduced computational resources affect inference time and object detection accuracy for CNNs on edge devices?

**RQ3:** How does limited memory capacity on edge devices influence the model inference speed and object detection accuracy of CNNs?

### 1.6.2 Hypotheses

**H1:** A CNN (like YOLO) can maintain at least 80% detection accuracy in low-light conditions on edge devices.

**H2:** Reduced compute resources increase inference time by 40%, leading to delayed object detection but < 10% decrease in accuracy.

**H3:** Reducing image resolution by 50% improves inference speed by 20% for memory capacity-limited edge devices without reducing object detection accuracy by more than 10%.

## 1.7 Scope of Research

The scope of research for this praxis includes several components necessary to test the implementation of a Convolutional Neural Network (CNN) model to provide real-time detection and identification of adversary military assets on resource limited edge devices. These components include a CNN model trained on various publicly available imagery data of modern Russian military armored ground assets. The scope of research for this Praxis also includes the test and evaluation of the CNN model leveraging a simulated environment to evaluate the effectiveness of the model in real-time detection under different real-world lighting conditions when deployed on an edge device. The objective of this research is to deliver deep learning capabilities on low-cost, scalable, and limited-compute edge devices capable of operating in denied, degraded, intermittent, or limited (DDIL) communication environments enabling real-time detection in alignment with the U.S. Department of Defense Replicator Initiative.

## 1.8 Research Limitations

Several research limitations in this praxis include quantity and quality of available image datasets of modern Russian military assets as well as test environments sufficient to evaluate CNN performance on sUAS devices. One of the primary image datasets of modern Russian military vehicles in the praxis contains images of military assets in a variety of conditions ranging from well-defined and recognizable to heavily damaged or unrecognizable prior to the implementation of data cleaning methods on the dataset. Some of these limitations introduce challenges to model training and validation including challenges with imbalanced classes, label consistency across datasets, and limited pre-defined bounding box data. Lastly, evaluation of the deep learning model is conducted in a simulated environment due to limitations in testing the model in real-world conditions.

## 1.9 Organization of Praxis

This praxis consists of five chapters that outline all steps conducted in the research and development of the proposed solution. Chapter 1 contains an introduction to the background, research motivations, scope of research, thesis, research questions, and hypotheses. Chapter 2 includes the literature review of related academic research pertinent to the development of this praxis to include non-government organizations (NGO), government, and military publications, as well as academic articles relevant to AI engineering. Chapter 3 explains the methodology used to research, develop, and test a lightweight CNN model. Chapter 4 investigates the research questions, documents the experiment results, and evaluates the hypotheses of this praxis. Chapter 5 outlines the analysis of the results and recommends potential areas for future research.

# Chapter 2—Literature Review

## 2.1 Introduction

The literature review for this praxis explores publications focused on key areas of research related to the use of artificial intelligence on small, unmanned aircraft systems (sUAS). This review explores publications from several sources including U.S. Department of Defense, U.S. Congress, non-government organizations (NGO), and academic peer-reviewed articles. These publications provide a comprehensive series of references on current capabilities and technologies. Additionally, the literature referenced in this praxis identifies strategic and tactical military requirements for the use of sUAS in modern conflicts.

Section 2.2 of this chapter introduces electronic warfare (EW) and its impact on sUAS. Additionally, this section primarily explores military government, and NGO publications on the applications of this technology in military conflict in Eastern Europe.

Section 2.3 explores the use of artificial intelligence on unmanned aerial vehicles (UAV). This section details applications for real-time object detection on UAVs. This section also identifies several capabilities pertinent to this praxis, like navigation and autonomous flight.

Section 2.4 of the literature review introduces the You Only Look Once (YOLO) object detection algorithm. This section explores several applications of the YOLO algorithm on sUAS platforms.

Section 2.5 of this chapter further explores the YOLO algorithm. It identifies training techniques and fine-tuning methods that can enhance the performance of the

model. This section also investigates several limitations of edge devices as well as considerations when implementing an object detection model on sUAS. Moreover, this section introduces the potential uses for simulated environments. These simulated environments may provide a controlled environment to test the detection performance of a deep learning model.

## 2.2 Electronic Warfare (EW) and Small Unmanned Aircraft Systems (sUAS)

Electronic warfare (EW) is a military technology used to disrupt electromagnetic signals (Kallenborn & Plichta, 2024). According to Molloy (2024), the use of EW in military conflicts has been successful in "defeating the current generation of UAS". EW can serve as a countermeasure to sUAS by interfering with the transmission and reception of communications between a drone and its operator (Kallenborn & Plichta, 2024). Publications from the U.S. Department of Defense (DOD), U.S. Congressional Research Service, Defense Systems Information Analysis Center (DSIAC), the Royal United Services Institute (RUSI), and the Rand Corporation, demonstrate key considerations pertaining to the utilization, prevalence, issues, and roles that EW and sUAS have in modern military contexts as well as key considerations and concerns in the proliferation of these devices (Gerstein & Leidy, 2024; Milonas et al., 2024; Sayler, 2024; Watling & Reynolds, 2023; U.S. Department of Defense, 2023). The purpose of research into EW and sUAS is to provide a greater level of context, data, and examples of areas where artificial intelligence and machine learning may provide solutions to key strategic issues related to military application of these technologies (Kallenborn & Plichta, 2024; Pledger, 2021; U.S. Department of Defense, 2024).

### 2.2.1 The Role of EW and sUAS in Modern Military Applications

The prevalence of sUAS in military applications has undergone significant advancements in miniaturization and capabilities over the past decade and have enabled a rapid increase in the deployment of sUAS devices in military applications in several modern conflicts (Molloy, 2024). Several authors, including Busch (2018), have chronicled how these technological advances have reduced barriers to applying sUAS devices in military conflicts at a rapid scale for various tasks including intelligence, surveillance, and reconnaissance (ISR). Additionally, the low cost of manufacturing and deploying sUAS has enabled smaller military forces with the potential capability to challenge or contest significantly larger military forces (Patterson, 2018). As identified by Hammes (2021), there have been significant benefits for smaller states when countering larger adversaries through the asymmetric and highly scalable nature of deploying sUAS devices in military applications.

### 2.2.2 Miniaturization and Capability Advancements of sUAS

Advances in the operational capabilities of sUAS has transformed these types of devices from experimental prototypes and consumer electronics into highly capable, low-cost, and rapidly scalable technology assets with broad military application (Finer & Shimer, 2025). Over the past several decades, improvements in payload, sensor capabilities, and onboard processing enable devices like sUAS to leverage an array of capabilities to include things like high resolution optical sensors, long-range encrypted radios for command-and-control, as well as increased flight endurance (Busch, 2018). In addition to this, Busch (2018) documents the impact that sUAS has on the increase of battlefield awareness in real-time across a myriad of devices and sensors traditionally reserved for larger, more complex reconnaissance systems.

Busch (2018) provides insights into the development and evolution of small drones and the increase in capabilities and reduction in size over the past several decades and documented several significant factors that have been proven useful in the military application of this technology. These factors include increased battlefield awareness and cost efficiency because of the low cost of acquisition and operation (Busch, 2018). Additionally, the deployment of multiple sUAS can provide rapid scaling of capabilities when implemented across large environments (Busch, 2018). Moreover, Busch (2018) identifies the effectiveness of EW countermeasures against sUAS. Busch (2018) describes the strategic implementation of EW countermeasures and how they are used to deny the transmission and reception capabilities of sUAS to impact the command-and-control (C2) of these devices.

### 2.2.3 Affordability and Tactical Impact in Contemporary Conflicts

The ease of access and affordability of sUAS has increased the use of the technology in modern military applications (Chávez, 2024; Sayler, 2024). According to Chávez (2024), sUAS provides militaries with adaptable and scalable capabilities in conflict environments. Several factors like automated flight controls, commercial availability and ease of use, made sUAS more prevalent on the modern battlefield (Chávez, 2024).

Chávez (2024), in the report *Small Drones for Big Militaries: The Way Ahead*, details the increased use of sUAS in modern conflicts and highlights the use of the technology throughout Eastern Europe following the Russian invasion of Ukraine in 2022. In this example, Russian and Ukrainian forces have leveraged the use of

inexpensive sUAS armed with explosive munitions to counter significantly more heavily equipped or expensive adversary assets (Chávez, 2024).

The increase in accessibility and affordability of modern sUAS has spurred the adoption and deployment of this technology by state and non-state actors in various military conflicts and applications (Chávez, 2024; Sayler, 2024). Chávez (2024) identifies the impact that the modular nature of sUAS has in providing adaptable and rapidly deployable military capabilities in various military conflicts. In particular, the use of commercially available flight controller software, minimal learning curve, and the ease of rapid procurement and deployment of sUAS devices has increased the prevalence of this technology on the modern battlefield (Chávez, 2024).

### 2.2.4 Policy and Strategic Framework for sUAS

The U.S. Federal Government and the U.S. Department of Defense (DOD) continue to underscore the strategic significance that small, unmanned aircraft systems (sUAS) and artificial intelligence augmented edge devices has on the development of national security policies and military applications (U.S. Department of Defense, 2023; Sayler, 2024). In the Congressional Research Service publication *Emerging Military Technologies: Background and Issues for Congress,* Sayler (2024) highlights various emerging technologies including the emergence and integration of artificial intelligence into lethal autonomous weapon systems (LAWS) as well as electromagnetic spectrum technologies. Artificial intelligence enabled devices such as sUAS can enhance reconnaissance, surveillance, and intelligence (ISR) capabilities as well as augment kinetic operations in highly contested environments (Sayler, 2024). Additionally, the low-cost, highly scalable nature of modern sUAS platforms may provide for the rapid

development and deployment of highly scalable sUAS capabilities augmenting military operations and meeting U.S. strategic interests (U.S. Department of Defense, 2023).

### 2.2.5 Electronic Warfare and Counter-Drone Defenses

The growing prevalence of sUAS in military applications by state and non-state actors has spurred an increase in the development of counter sUAS (C-sUAS), EW systems, and technologies designed with various capabilities to neutralize small drones (Kallenborn & Plichta, 2024). In the publication *Breaking the Shield: Countering Drone Defenses*, Kallenborn and Plichta (2024) provide insightful analysis of multi-sensor arrays used to detect, track, and neutralize sUAS devices through the combination of electromagnetic spectrum scanning and electro-optical sensors. Once identified, sUAS devices may be neutralized through the application of EW to jam, spoof, or otherwise counter the electromagnetic signals necessary for command-and-control functions of the sUAS device (Kallenborn & Plichta, 2024). According to Kallenborn and Plichta (2024), there are several technologies used to counter against sUAS (c-sUAS). Technologies like radar, radio frequency detection, acoustic sensors, and visual detection and identification are used in the application of C-sUAS (Kallenborn & Plichta, 2024).

### 2.2.6 Electronic Warfare in the Russia-Ukraine Conflict

Watling and Reynolds (2023) analyze Russian military strategy to include the prevalence of EW as a countermeasure used to defeat sUAS. According to Watling and Reynolds (2023), in *Meatgrinder: Russian Tactics in the Second Year of Its Invasion of Ukraine*, the Russian military successfully implemented EW to counter Ukrainian drones after following its invasion of Ukraine in 2022. Watling and Reynolds (2023) document the increased use of EW by the Russian military throughout the war in Ukraine to disrupt

signals of Ukrainian sUAS. As a result, during the first two years of the war in Ukraine, the use of EW by the Russian military caused a loss of ninety percent of all Ukrainian sUAS (Watling & Reynolds, 2023). The use of EW as a sUAS countermeasure disrupts intelligence, reconnaissance, and surveillance (ISR), command-and-control (C2), and secure communication through jamming and decryption of these signals. (Watling & Reynolds, 2023).

### 2.2.7 Asymmetric Advantages of AI Augmented Drones

Hammes (2021) provides insight into emerging technologies that are shaping modern defensive strategies, including through the proliferation of sUAS devices and autonomous defense systems. Hammes (2021) highlights an advantage that small countries may be able to leverage through the implementation of small low-cost defense technologies in their military arsenals to counter the capabilities of stronger neighboring nations. Hammes (2021) identifies the significant impact that low-cost drones and artificial intelligence enabled sensors provide in the defensive capabilities of small states and highlights several key factors that these technologies provide to equalize the defensive terrain of small nations from offensive capabilities of potential adversaries with greater traditional military capabilities. Technologies such as swarm and artificial intelligence augmented drones may be affordable and rapidly scalable low-cost defense systems that can be leveraged for intelligence, reconnaissance, and surveillance (ISR) as well as kinetic capabilities (Hammes, 2021).

Hammes' (2021) publication, *Defense Dominance: Advantages for Small States* provides useful insights and context into the challenges and application of AI enabled sUAS platforms. Moreover, the ideas presented by Hammes (2021) conceptually align

with the strategic deployment of artificial intelligence augmented low-cost sUAS devices requested by the U.S. DOD Defense Innovation Unit (DIU) Replicator Initiative. The key conceptual overlaps between Hammes' insights and the Replicator initiative provide validation of the use case of this praxis.

#### 2.2.8 Security Threats Facing Drones

Samanth et al. (2022) provides key insights into security considerations of drones as connected edge devices in the article *Security in Internet of Drones: Threats and Countermeasures*. Samanth et al. (2022) defines several areas of security threats facing unmanned aerial vehicles operating with wireless communication to an operator or cloud-based computer resources. Samanth et al. (2022) identifies physical layer attacks against drones as the first security threat and includes attacks such as the jamming or spoofing of GPS signals to the drone, interference with sensors, or the physical incapacitation of the drone. Additionally, Samanth et al. (2022) describes how network application layer attacks like denial of service and man in the middle attacks could be used to disrupt or intercept the control of a drone. The security threats identified by Samanth et al. (2022) provide support in the research of the impact of EW on sUAS for this praxis.

### 2.3 AI/ML and sUAS

#### 2.3.1 Convolutional Neural Networks on Edge Devices

Research into the use of convolutional neural networks (CNN) on sUAS may potentially provide a solution to mitigate the vulnerabilities stemming from EW, particularly the loss of command-and-control capability. According to De Oliveira Barbosa et al. (2025), CNNs are a type of deep learning model that is made up of several layers. These layers process information in a forward feed that allows an artificial

intelligence model to extract features from data (De Oliveira Barbosa et al., 2025). Additionally, lightweight CNN models may be able to run on small edge devices with the need for external connectivity (Lee et al., 2024; Vaddi et al., 2019).

### 2.3.2 Artificial Intelligence Based Control for Drones

Manikandan et al. (2024) identifies the potential use of artificial intelligence on UAVs to provide various tasks that could support military and security applications. Manikandan et al. (2024) highlights key areas of research and context surrounding the application of convolutional neural networks (CNNs) for real-time object detection and identification on drones for object avoidance, image recognition, and non-GPS image-based navigation as well as use cases including challenges to GPS and remote piloted drones in EW disrupted environments (to include GPS jamming and spoofing).

Devos et al. (2018) provides insights into the development of a computer vision neural network for real-time obstacle avoidance for autonomous navigation applications on drones in the article *Development of Autonomous Drones for Adaptive Obstacle Avoidance in Real-World Environments*. According to Devos et al. (2018) there are several key considerations in the development of an autonomous navigation model including the reliance on high computational power, hardware, and sensors with conventional computer vision methods as well as the impact that these had with the deployment of smaller drones. Additionally, Devos et al. (2018) and Ma et al. (2023) identify that hardware-based flight controllers like the Pixhawk platform may potentially be useful to implement autonomous flight and navigation, further expanding automated capabilities of drones. Similarly, in the article *AI Wings: An AIoT Drone System for Commanding ArduPilot UAVs,* Lai et al. (2023) explores the possible application of

artificial intelligence paired with ArduPilot software to perform real-time automated flight control of small drones. In addition to these authors, Arshad et al. (2022) provides research into the application of using CNNs for the autonomous navigation of UAVs through complex environments. The valuable insights of Arshad et al. (2022) into the development and optimization of a CNN for drone navigation highlights several key insights into various challenges and considerations that impact the success of autonomous visual navigation in complex visual environments. The research of these authors provides insight into the potential application of artificial intelligence on UAVs for autonomous flight and provides context for this praxis.

### 2.3.3 YOLOv4-Tiny for Real-Time Detection on Edge Devices

In the article, *A Study on the Advancement of Intelligent Military Drones: Focusing on Reconnaissance Operations,* Lee et al. (2024) describes the use of a YOLOv4 based CNN to conduct real-time object detection on drones. The results of the research of Lee et al. (2024) demonstrate strong results with model confidence levels above ninety percent with real-time inference speeds. The demonstration by Lee et al. (2024) of deploying a CNN on an edge device provides valuable information into the requirements needed to successfully implement object detection on a UAV platform. Additionally, the challenges and constraints identified by Lee et al. (2024) are useful to the development of this praxis. Lee et al. (2024) identifies things like complex visual environments, hardware limitations, and flying altitude as key considerations when implementing real-time object detection on UAVs. Lastly, Lee et al. (2024) identifies the YOLOv4-tiny architecture as a potentially viable model for UAV based deployments

because of its compatibility with limited resource embedded systems and provides insight into relevant architectures for UAV deployed artificial intelligence applications.

### 2.3.4 CNN Object Detection Features and Optimizations on UAVs

Vaddi et al. (2021) analyzes various CNN architectures and how they perform with UAVs. According to Vaddi et al. (2021), CNNs like ResNet and MobileNet may be able to provide object detection capabilities on UAVs that have limited computational resources. The authors also identify common challenges with implementing CNNs on UAVs like complex environments, distant objects, and poor visibility (Vaddi et al., 2021). Z. Zhang et al. (2022) expands on these challenges and highlights how factors like memory, computational, and power constraints can impact model performance by increasing inference times.

### 2.3.5 Evaluation of Deep Learning Models on UAVs

Micheal et al. (2020) explores the use of deep learning models on UAVs. The authors identify several factors like altitude, occlusion, and perspective that may affect the performance of deep learning models on a UAV (Micheal et al., 2020). Kumar and Jannesari (2023) in *Optimal Deep Learning model for UAVs: A Case Study*, explore the deployments of different deep learning models on UAVs to detect and navigate trails. Kumar and Jannesari (2023) develop several models offline for implementation on drones for real-time testing. Kumar and Jannesari (2023) identify key considerations in the deployment of computer vision on edge devices to include computational and memory constraints, complex environments, visibility and variations in lighting, as well as the relationship between model accuracy and computational speed for real-time autonomous navigation. The results of Kumar and Jannesari (2023) demonstrate the

feasibility of leveraging a CNN like MobileNet to conduct complex classification tasks found in visual navigation on resource limited hardware like the NVIDIA Jetson Nano TX2. These findings indicate the feasibility of CNNs for use on computationally and memory limited edge devices with consideration to power consumption and object detection and inference speed (Kumar & Jannesari, 2023).

### 2.3.6 Small Scale Object Detection for UAVs

Saeed et al. (2023) provides insight into the use of CenterNet for object detection specifically with unmanned aerial vehicle (UAV) applications in the article *On-Board Small-Scale Object Detection for Unmanned Aerial Vehicles.* Saeed et al. (2023) leverages several datasets and CNN architectures optimized for edge devices with limited computational and memory capabilities to identify several major considerations and limitations in image detection of small objects, image detection of objects in complex or dense imagery environments, limitations in image detection due to low resolution and environmental factors such as lighting and occlusion, as well as computational and memory limitations. Saeed et al. (2023) identifies that the optimization of the CenterNet architecture may aid in these limitations and result in the implementation of techniques such as cascade corner pooling and center pooling to improve contextual information for images. Additionally, Saeed et al. (2023) identifies the efficacy for deployment of CNN models for use on NVIDIA Jetson Nano hardware for UAV applications.

### 2.3.7 Resolution and Power Consumption Trade-Offs for Image Detection

Abubaker et al. (2023) provides key insights into several factors and constraints that may impact the feasibility of drones for image collection in the article *Optimal Altitude Range for Image Collection Drone from Agricultural Areas with Image*

*Resolution and Power Consumption Constraints.* The article describes the impact that image resolution, power consumption, and altitude have on the effectiveness of image collection from UAVs (Abubakar et al., 2023). Although not applied in a military application, these factors may also provide key context into major considerations and constraints faced by defense applications of small drones in contested environments.

#### 2.3.8 Real-Time Object Detection of Vehicles on Edge Devices with AI

Shaikh and Gajjar (2023) provide key insights into real-time object detection using iterations of the YOLO computer vision deep learning models applied on edge devices with the goal of leveraging CNN-based object detection to detect and identify ambulances in real-time. The article *Performance Evolution of Ambulance Detection and Classification Using Machine Learning Through a Drone on Edge AI Devices* identifies key challenges, considerations, and efficacy when deploying the YOLO object detection architecture on a resource limited edge device like the Nvidia Jetson Nano (Shaikh & Gajjar, 2023).

### 2.4 You Only Look Once (YOLO)

The introduction of the You Only Look Once (YOLO) object detection model by Redmon et al. (2016) demonstrates a novel approach to the implementation of a detection algorithm built on a single stage detector. This algorithm uses classifiers to detect objects in real-time using a single CNN (Redmon et al., 2016). The model uses regression to find the location of an object in an image by splitting the image into a grid, predicting a confidence score based on the presence of a trained image class within each grid square, and generating a bounding box that encompasses the entire detected object (Redmon et al., 2016). According to Redmon et al. (2016), YOLO uses twenty-four convolutional

layers to identify features of an object. After the features are identified, two fully connected layers are then able to predict an object's location in an image (Redmon et al., 2016).

### 2.4.1 YOLO Model Architectures

Throughout the literature review of this praxis, several variations of the YOLO architecture have been identified. These variants include standard iterations like YOLOv4, YOLOv5, and YOLOv8 as well as modified iterations like G-YOLO for use on UAVs (X. Zhao et al., 2024). According to X. Zhao et al. (2024), YOLO variants like YOLOv8 can provide strong detection performance while maintaining fast inference times, which may be particularly suitable for UAV applications. According to Zhan et al. (2021) in *An improved YOLOv5 real-time detection method for small objects captured by UAV,* the implementation of image detection capabilities on UAVs provides several challenges that include degraded performance of the YOLOv5 algorithm with drone captured images versus traditional forms of image capture. Zhan et al. (2021) identifies that small image size is one of the primary culprits for the degraded model performance onboard a UAV platform. Some of the key insights by Zhan et al. (2021) into the improvement of small object detection is the identification of three techniques including the modification of the model's network, augmentation of the model's data, and through the implementation of Generative Adversarial Networks (GAN).

### 2.4.2 Robust Real-Time Object Detection with YOLO Architecture

Parmar et al. (2023) provides key insights into the application of computer vision models on edge devices to identify potholes and adverse road conditions in the article *Drone-Based Potholes Detection Using Machine Learning on Various Edge AI Devices*

*in Real-Time*. Parmer et al. (2023) introduces a framework to capture and process image data in real time to detect and identify potholes and provides key insights into the suitability of various computer vision models including iterations of the YOLO architecture. Parmer et al. (2023) shows the effectiveness of deploying the YOLO architecture on various edge devices including the Raspberry Pi 4 and NVIDIA Jetson Nano with varying levels of detection accuracy ranging from approximately 70% up to 96% when using YOLOv5, YOLOv7, and YOLOv8 while maintaining high inference speed necessary for real-time detection.

Wu et al. (2024) introduces a novel method of leveraging real-time object detection with the YOLOv5 computer vision model to detect stray dogs and transmit the location of the dog to a separate device. Although not in direct alignment with the military application of image detection on UAV platforms, Wu et al. (2024) provides insights of leveraging the YOLO architecture to implement real-time object detection in visually complex environments. Wu et al. (2024) also provides insight into the optimization of accuracy and inference time for deployment on edge devices, as well as data selection and training methodologies that may provide insight into similar constraints and limitations found in military UAV applications.

### 2.4.3 YOLO for Unmanned Aerial Vehicle Deployment

Ding et al. (2024) demonstrates the effectiveness of the YOLOv8 architecture for infrared object detection on unmanned aerial vehicles (UAV) and provides insights into a lightweight model tailored to real-time infrared object detection for use in unmanned aerial vehicle applications. In the article *A Lightweight Real-time Infrared Object Detection Model Based on YOLOv8 for Unmanned Aerial Vehicles* Ding et al. (2024)

introduces the Lightweight Real-Time Infrared YOLO (LRI-YOLO) which is built off the You Only Look Once (YOLO) framework. According to Ding et al. (2024), there are several challenges and considerations that stem from infrared image capture like low image resolution and small object size which may severely impact the model's performance. Additionally, Ding et al. (2024) identifies the challenge that computational and memory constraints have on the implementation of object detection models on unmanned aerial vehicles.

In the article *High-Precision Detection for Sandalwood Trees via Improved YOLOv5s and StyleGAN,* Y. Zhang et al. (2024) uses YOLOv5 to remotely detect sandalwood trees. According to Y. Zhang et al. (2024), a lightweight object detection model like YOLOv5 may be adequate to identify sandalwood trees from other trees in a forest. Y. Zhang et al. (2024) also identifies several factors that could influence the model's performance on an edge device. These include things like visually complex environments, limited data for training, and hardware limitations on edge devices (Y. Zhang et al., 2024).

Similar factors are also identified by Han et al. (2020). The authors identify that the use of a lightweight CNN like YOLO may be impacted by hardware limitations. To demonstrate this point, Han et al. (2020) deploys a YOLO model on an NVIDIA Jetson computer, highlighting the benefits and limitations of dedicated hardware for edge deployments of artificial intelligence. Another use case of YOLO on a UAV platform is demonstrated by Hakani and Rawat (2024). The authors identify how the increased use of UAVs has introduced new security concerns to critical infrastructure (Hakani & Rawat, 2024). In order to mitigate these threats, Hakani and Rawat (2024) implement a YOLO

based detection model on an NVIDIA Jetson computer for real-time detection of drones. Alhassan and Yilmaz (2025) identify several variables and considerations for the successful deployment of edge computing hardware to UAVs including limitations to battery capacity, weight constraints, and computational limitations. To accommodate this, Alhassan and Yılmaz (2025) demonstrate the application of the YOLOv4 and YOLOv5 algorithms for image detection on various edge computer platforms including the NVIDIA Jetson Nano in the article *Evaluating YOLOv4 and YOLOv5 for enhanced object detection in UAV based surveillance.*

### 2.4.4 YOLOv8 Selection for Edge Applications

The prevalence of electronic warfare (EW) on modern battlefields and contested military environments heavily impacts the efficacy of remotely operated drones due to the impact that EW has in denying command-and-control, telemetry, and other communication capabilities (Watling and Reynolds, 2023). According to Samanth, et al. (2022) and Gerstein and Leidy (2024), computer vision based autonomous systems may provide unmanned aerial vehicles (UAV) with the capability to continue operation in otherwise denied, degraded, intermittent, or limited (DDIL) communication environments. The increased capabilities of edge-based computing paired with the advancement of deep learning computer vision models may provide the potential mitigation of the EW threat facing UAVs in contested environments enabling semi-autonomous enhancements to remotely operated devices.

In particular, the implementation of deep learning computer vision models may provide a sufficient level of capability with generalization when detecting and identifying objects. As identified by Ma et al. (2023) and Dou et al. (2025), deep learning models

like the YOLO architecture may provide adequate generalization in complex or visually cluttered environments. The benefits of deep learning computer vision models in this application may potentially extend to mitigate EW impacts for vision tasks on drones in contested environments to include detection and identification of adversary assets as well as vision-based navigation and obstacle avoidance.

However, the capabilities and benefits of the application of deep learning computer vision on UAVs may only be beneficial if there is adequate processing capability on the edge device to provide near real-time processing within the computational and power limitations of the edge device. Recent developments over the past several years have provided increased computational and power efficiency with low-power platforms like the NVIDIA Jetson Nano and NVIDIA Jetson Orin Nano which provide adequate processing capability and power efficiency enough to support the application of quantized models (Hakani & Rawat, 2024; Han et al., 2020; Mittal, 2024). The combination of advancements in deep learning computer vision models and advancements in edge computing hardware may provide an adequate environment for the application of deep learning object detection on edge devices equipped to unmanned aerial vehicles deployed in EW contested environments.

According to Terven et al. (2023), there are various common applications suitable for the deployment of the YOLO algorithm for real-time object detection to include driverless automation, advanced robotics, and video surveillance applications. In the paper *A Comprehensive Review of YOLO architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS*, Terven et al. (2023) documents the robust nature of the YOLO algorithm since the introduction of it in 2016. Introduced in 2023 by

Ultralytics, YOLOv8 is an updated version of the YOLO algorithm that was available in five different scaled versions to accommodate deployments to different applications ranging in size from nano, small, medium, large, and extra-large formats (Terven et al. 2023). According to Terven et al. (2023), the improvements to the YOLOv8 algorithm include improvements to the backbone architecture that resulted in improved detection performance.

Yao et al. (2024) highlights the fundamentals of the YOLOv8 model and its strong balance between performance and speed. The YOLOv8 model is comprised of three primary parts: a backbone used for feature extraction that leverages the C2f module, a head used for the separation of classification and localization, and a neck that leverages the Path Aggregation-Feature Pyramid Network (PA-FPN) (Yao, et al, 2024). Although these components standard to the YOLOv8 model demonstrated strong overall performance, according to Yao et al. (2024) the standard model faced challenges in the detection of small objects, many which stemmed from visually complex environments which led decreased detection.

Sanikommu et al. (2025) demonstrates one of the potential applications for the YOLO algorithm in maritime environments on edge devices for the detection of ships and ship ports. Sanikommu et al. (2025) identifies the application of an NVIDIA Jetson Nano edge computer to provide maritime surveillance capabilities in remote environments with computational, signal, and power limitations. To enhance the accuracy and performance of the model, Sanikommu et al. (2025) identifies and applies various data augmentations during training to increase the robustness of the YOLO model. Techniques included various image modifications such as flipping the image horizontally

and vertically, random scaling and cropping of the image, as well as rotation of the image (Sanikommu et al., 2025). These techniques in particular aided Sanikommu et al. (2025) in the mitigation of class imbalance and enhancing model robustness. Moreover, the concept of image modification techniques was also demonstrated by Dou et al. (2025) with data augmentation techniques such as random image rotation, flipping, scaling, and cropping as well as random adjustments to color, contrast, saturation, brightness, hue, and the introduction of noise to images.

Saydirasulovich et al. (2023) demonstrates the improvement and modification of the YOLOv8 algorithm to recognize smoke detection leveraging UAV images across various weather and lighting conditions. Although a modification of the YOLO model's network was implemented, several model training optimization techniques and challenges are consistent with those of the standard YOLOv8 model to include the requirement for substantial and well-labeled data as well as the mitigation of challenges such as overfitting, underfitting, imbalanced classes, and data-limitations (Saydirasulovich et al., 2023).

The YOLO (You Only Look Once) architecture was one of the most prevalent deep learning architectures identified during the research phase of this praxis. The work of various authors, like Ali and Zhang (2024) and Terven, et al. (2023) provide insight into the architecture's capabilities for providing real-time object detection with consistent accuracy. According to Redmon, et al. (2016), the YOLO architecture can perform classification and localization in a single stage that enables reduced inference time when compared to other vision models that rely on two stages for detection. The reduction in inference time during detection tasks may provide an increased level of benefit in the

application of deep learning computer vision on unmanned aerial vehicles where real-time inference is required.

## 2.5 Training, Optimization, Simulation

### 2.5.1 Optimization of YOLOv8 to Minimize Computational Overhead

X. Zhao et al. (2024) demonstrates the efficient detection of small objects using YOLOv8. The authors identify important factors like complex visual environments and limited computational resources that could impact the model's performance (X. Zhao et al., 2024). In addition to identifying similar resource constraints, De Oliveira Barbosa et al. (2025) uses several optimizations to increase the performance of a YOLOv8m model on an NVIDIA Jetson computer. The authors use techniques like image rotation, image flipping, as well as adjustment to brightness and contrast to improve the results of the model's training (De Oliveira Barbosa et al., 2025). The author's selection of the medium size version of the YOLOv8 algorithm was justified by De Oliveira Barbosa et al. (2025) as it provided strong performance with real-time image detection applications.

### 2.5.2 Small Object Detection with YOLO

According to Li and Wu (2024), there are several challenges for real-time object detection applications from UAV platforms due to the nature of image collection from an elevated platform. Some of these challenges include small image size due to UAV altitude as well as variance in angle and perspective which could impact a detection model's performance (Li & Wu, 2024). Li and Wu (2024) introduce a modified version of the YOLOv8 algorithm that included modifications to the neck structure of the architecture to successfully improve identification of features necessary for small object

detection and demonstrates increased levels of performance in comparison with the standard YOLOv8 algorithm.

According to Qu et al. (2025) and Terven et al. (2023), object detection algorithms like YOLOv8 have extensive uses across several different industries including military applications but may exhibit limitations with small object detection. Qu et al. (2025) identifies that the YOLOv8 algorithm faced challenges in the detection of small objects which impacted the performance of the deep learning algorithm specifically with UAV collected image data. To remedy this, Qu et al. (2025) implements several modifications to the algorithm's network architecture to limit parameters and improve detection of small objects over the standard version of YOLOv8.

Zhai et al. (2023) identifies limitations of the YOLOv8 algorithm when detecting small objects from UAV platforms. Object detection on UAVs require several considerations including model accuracy and real-time detection (Zhai et al., 2023). Moreover, Deng et al. (2023) similarly identifies the limitations of the YOLO algorithm in detecting small objects from UAVs. According to Zhai et al. (2023), factors like image complexity, object size, and lighting conditions may potentially impact the performance of a YOLOv8 model. Moreover, Rey et al. (2025) identifies additional factors like computational limitations that may impact a YOLOv8 model's effectiveness on an edge device.

### 2.5.3 Data Generation with AirSim and Unreal Engine

In the article, *Analyzing the Efficacy of Synthetic Images in Unmanned Aerial Vehicle Detection* by Arezoomandan et al. (2024), the authors use Microsoft's AirSim software to augment real-world datasets with computer generated image data using

Unreal Engine. In addition to this, Arezoomandan et al. (2024) identifies how synthetic data could be used to augment existing datasets. Because of this, it may be beneficial for this praxis to implement a simulated test environment like the Unreal Engine. Additionally, factors like weather and lighting could be used to test an object detection model safely in a virtual environment (Arezoomandan et al., 2024). The article *Analyzing Deep Reinforcement Learning with Shapley Values,* Çetin et al. (2023) explains the feasibility of using AirSim to test deep learning models as well.

### 2.5.4 Simulated Test Environments using Game Engines

Farkhodov et al. (2023) provides key insights into the research, development, and testing of the application of Deep Reinforcement Learning (DRL) with TensorFlow agents to train drones to autonomously track moving objects using Microsoft AirSim in the article *Deep Reinforcement Learning TF Agent-based Object Tracking with Virtual Autonomous Drone in a Game Engine*. Farkhodov et al. (2023) examines the use of a virtual test environment to experiment with deep learning models in the detection of UAVs. Farkhodov et al. (2023) uses AirSim paired with the Unity game engine for the development of a virtual test environment to test UAV object tracking capabilities. The successful demonstration of AirSim on top of the Unity game engine as a practical and robust virtual testing platform for the development of autonomous capabilities on UAVs is significant in the development and selection of the testing methodology in this praxis.

In addition to the implementation of AirSim for UAV object tracking, Xie et al. (2023) provides insights into the development and implementation of a testing framework to evaluate route planning and monitoring for UAVs across landslide prone environments with a simulated environment. Xie et al. (2023) demonstrates the utility of leveraging

AirSim built on the Unreal Engine to provide photorealistic simulated environments to test UAV monitoring capabilities. Xie et al. (2023) provides insights into key considerations and limitations when implementing simulated test environments.

According to Shimada et al. (2022) in *Pix2Pix-based Monocular Depth Estimation for Drones with Optical Flow on AirSim,* simulation software may provide an adequate virtual environment to test obstacle avoidance and navigation on autonomous drones. Shimada et al. (2022) leverages AirSim as a simulated environment to test collision detection and avoidance with drones within a controlled environment through the implementation of several strategies during testing to include the simulation varying conditions like altitude, drone rotation, and lighting. The results identified by Shimada et al. (2022) demonstrate how simulated environments may be suitable for controlled testing of real-time object detection.

**2.6 Analysis**

The articles referenced provide important insights into the potential application of a lightweight CNN on sUAS for military applications. The various use cases identified in this chapter highlight factors that may aid in developing an artificial intelligence model to mitigate the impacts of EW on sUAS.

Several of the sources in the literature review of this praxis leverage lightweight deep learning models based on the YOLO architecture to maximize the effectiveness and efficiency of computer vision on edge devices with memory, processing, and power constraints. The efficiency of the one stage detector of the YOLO architecture, variety of model sizes, image processing capabilities, and real-time image inference speeds provide

evidence to support the efficacy of a CNN like YOLO for object detection applications on edge devices like sUAS platforms.

Additionally, various applications the YOLO architecture referenced in this literature review ranging from ambulance detection, pothole detection, stray dog detection, as well as others, demonstrate the strong balance between detection accuracy and real-time inference speeds of the architecture. The use of these models on resource constrained hardware further demonstrates the relevance of this research in the development of a deep learning model for the detection of adversary military assets from an sUAS platform.

In addition to the findings identified in the literature review pertaining to the implementation of deep learning models such as the YOLO machine learning architecture, several key insights into methodologies and platforms to develop and test object detection and identification by sUAS are identified. Several academic articles referenced during the research portion of this praxis are used in the identification of testing methodologies for computer vision deep learning models on small drones. In particular, AirSim is referenced in several academic articles cited in this praxis. Several articles provide key insights into the use and efficacy of leveraging a simulated three-dimensional environment to test various deep learning capabilities on drones within a controlled environment. Additionally, many of the referenced articles provide insights and considerations pertaining to the configuration of experimentation conducted in a simulated environment which influenced the test environment used in this praxis.

# Chapter 3—Methodology

## 3.1 Introduction

This chapter addresses the methods used to develop research questions and test the hypotheses in this praxis. The methodology outlined in this chapter suggests a potential solution that may help reduce the threat of electronic warfare (EW) on small, unmanned aircraft systems (sUAS) in disrupted environments. The research in this chapter identifies several important factors that may aid in the development of a deep learning artificial intelligence model made for edge devices with limited resources. There are several initial steps to establish the initial baseline programming environment needed to train, test, and optimize a CNN for the use case identified in this praxis. These steps include data source selection, data cleaning, model selection, model training, as well as selection of a test environment.

## 3.2 Research Design and Approach

The research in this praxis is conducted in several phases to develop a holistic approach to identifying an issue within the defense sector, understanding the environment in which the issue exists, identifying consequences of the issue, and compiling research across various forms of media to include military and government publications as well as industry peer reviewed research. The publications referenced in developing this praxis primarily fall into two categories: scientific peer-reviewed publications pertaining to the engineering of artificial intelligence and government or military publications pertaining to electronic warfare, drones, technology strategy, as well as research of future defense issues. The combination of these sources provides the necessary insight into the nexus

between current artificial intelligence engineering research, existing applications, and the implementation of artificial intelligence on low cost and scalable edge devices within the defense sector.

### 3.3 Prototyping

There are several datasets used in the research, prototyping, and selection of machine learning frameworks in the development of this praxis. In the initial research and prototype phase of the praxis, several small image datasets are tested to determine the basic functionality of a prototype code environment necessary to conduct initial experimentation with a deep learning computer vision model. These included the CIFAR-10, MNIST, and COCO Tiny datasets as well as a small sample of the *Oryx 2022 Russia Ukraine War* dataset. A small subset of the *Oryx 2022 Russia Ukraine War* dataset is used for initial testing of the machine learning frameworks to verify functionality of the code environment as well as compatibility and usability of the vehicle image dataset. The full *Oryx 2022 Russia Ukraine War* dataset includes approximately 20,000 images of military vehicles across 25 categories to include Russian main battle tanks (MBT), armored personnel carriers (APC), infantry fighting vehicles (IFV), various light armored vehicles (LAV), aircraft, artillery, as well as several other categories of military assets. For the initial prototype testing of the code environment, 100 images from the tank, armored personnel carrier, and infantry fighting vehicle images are randomly selected. Several shortcomings are identified with the *Oryx 2022 Russia Ukraine War* dataset including a lack of bounding box data for the labeled images. To mitigate this issue, initial bounding box data is generated manually for the sample of the dataset using Roboflow to draw the boxes and export the bounding box data.

### 3.4 Data Sources

To accomplish the training and evaluation of a CNN model tailored for edge devices used in this praxis, it is necessary to collect an adequate amount of labeled image data of Russian military vehicles. There are three primary datasets leveraged in the development of this praxis to provide an adequate amount of image data, classes, and features necessary for the CNN model to generalize well against unseen image data within a simulated environment. The publicly available data sources used in this project included the *Oryx 2022 Russia Ukraine War* dataset which consisted of approximately 20,000 images of primarily Russian military assets, the *War_Tech 2.0* image dataset which consist of approximately 2,500 images of primarily Russian military assets, as well as the *Military_Vehicles* image dataset which consisted of approximately 8,000 images of primarily Russian military assets.

Each of the datasets provide image data of Russian military assets categorized into several categories including specific models of vehicles or overarching vehicle categories such as tanks, armored personnel carriers, and others. The use of all three datasets is leveraged due to the need for an adequate selection of images to provide enough features to enable the model to generalize well against different environments, lighting conditions, and levels of image obfuscation. The *Oryx 2022 Russia Ukraine War* dataset is made of recent images of a variety of Russian military assets. This collection of images from this dataset stem from Oryx, a Dutch open-source intelligence website recognized for its detailed coverage of the 2022 Russian invasion of Ukraine (Hambling, 2022). This dataset provides images of Russian military vehicles in complex modern

battlefield environments. For this reason, this dataset may be very useful to train a model for real-time object detection for military applications.

### 3.4.1 Data Cleaning and Structure

Before cleaning and preparing the data, exploratory data analysis (EDA) is performed on all three datasets to find key characteristics within the data. This includes visualization of samples of images and labels, as well as identification of the folder and class structure of the dataset. The *War_Tech 2.0* and *Military_Vehicles* datasets include label data with bounding boxes within a comma-separated value (CSV) format. The *Oryx 2022 Russia Ukraine War* dataset includes label data within a CSV format but without any bounding box data. Due to the significance of the recently collected images of Russian military assets within the *Oryx 2022 Russia Ukraine War* dataset, additional steps are required in the data treatment process to ensure compatibility with the YOLO architecture.

There is some inconsistency with the label naming conventions between the different datasets. For the purpose of this praxis, individual vehicle model names are collapsed into broader asset classes in order to group similar military capabilities together. This technique helps to ensure that there are enough vehicle images in each class that share similar visual characteristics. The general asset classes that are selected for this praxis include infantry fighting vehicles (IFV), armored personnel carriers (APC), main battle tanks (MBT), light armored cars (LAV_CAR), and light armored trucks (LAV_TRUCK). Each of these general asset classes contains images of various military vehicle models that each shared similar capability and visual characteristics as shown in Figure 1.

**Figure 1: Vehicle Classes**

Additional data cleaning steps are needed in order to use the *Oryx 2022 Russia Ukraine War* dataset since bounding box data is needed to train the YOLO model. To accomplish this, the *War_Tech 2.0* and the *Military_Vehicles* datasets are leveraged to train an initial YOLOv8x (extra-large) model on a desktop environment. This model is used to perform object detection on the *Oryx 2022 Russia Ukraine War* dataset as a method to identify and save the bounding boxes for the dataset.

### 3.4.2 Data Preparation

In this praxis there are several steps used to prepare the image data. These include the selection of datasets, the cleaning of the image data, and the preparation for model training. Images from the datasets are resized to 640x640 pixels and organized by class

label. After the images and labels are organized, images of military vehicles needed for this praxis are selected and copied over to a consolidated dataset. Additionally, the remaining data is further cleaned through the removal of heavily damaged or unrecognizable vehicle images. Next, the consolidated dataset is divided into three parts: seventy percent of the images for model training, twenty percent of the images for model validation, and ten percent for model testing. Finally, augmentation of the data using techniques like oversampling, cropping, rotation, and distortion are used to limit class imbalance and assist with model training.

## 3.5 Model Research and Selection

The research in this praxis identifies that the You Only Look Once (YOLO) architecture may be suitable for real-time object detection for sUAS. There are several considerations that were identified during the research portion of this praxis including model size, inference speed, and detection performance. The most prevalent architecture identified during the research phase of this praxis includes several YOLO (You Only Look Once) variants for applications that require real-time object detection and require lightweight feature extraction capability.

The YOLO architecture provides efficient object detection capabilities that may be suitable for use on resource constrained edge devices like sUAS. This in combination with the balance of performance and inference speed may make the YOLO architecture useful to potentially augment sUAS detection of adversary military vehicles.

### 3.5.1 YOLO Variants for Object Detection

This praxis references several authors that demonstrate several use cases where YOLO is applied to small drones for object detection. According to De Oliveira Barbosa

et al. (2025), YOLO provides accurate detections and fast inference speed necessary for real-time applications. Additionally, these characteristics paired with the model's limited computational requirements may make it suitable for use with UAVs (Ding et al., 2024; B. Zhao et al., 2024). B. Zhao et al. (2024) identifies the possible application of YOLO on UAV platforms for maritime search and rescue applications. To improve the detection of small objects from a UAV, B. Zhao et al. (2024) suggests several optimization methods to potentially enhance the model's detection performance. Furthermore, B. Zhao et al. (2024) also identifies several potential challenges for UAV deployed object detection including things like variations in weather, different camera angles, and object distance.

### 3.5.2 YOLOv8 Architecture

There are three main parts to the YOLOv8 architecture: a backbone, neck, and head (Wang et al., 2023). The model uses the CSPDarknet53 backbone which extracts features from images (Wang et al., 2023; Yao, et al, 2024; Zhai et al., 2023). The model includes a neck that uses the Path Aggregation-Feature Pyramid Network (PA-FPN) for feature fusion (Wang et al., 2023; Yao, et al, 2024; Zhai et al., 2023). Lastly, the model leverages an anchor-free head for regression, classification, and object detection (Wang et al., 2023; Yao, et al, 2024; Zhai et al., 2023). De Oliveira Barbosa et al. (2025) highlights that YOLOv8 provides a strong balance between detection accuracy and inference speed. Additionally, De Oliveira Barbosa et al. (2025) identifies that the YOLOv8m variant maintains strong performance with a more compact size suitable for edge devices.

## 3.6 Experiment Environment

The experiment environment includes several components such as multiple image datasets, a desktop coding environment for hosting the training of a lightweight YOLOv8 model, a small edge computer for the deployment and testing of a YOLOv8 model for computer vision applications on sUAS platforms, as well as a virtual environment for the generation of simulated image data in leu of a real-world test environment.

### 3.6.1 Desktop Coding Environment

This praxis leverages a desktop coding environment for data cleaning and model training. A desktop environment is selected for this purpose to ensure adequate resources are available to train and test extra-large and medium size YOLOv8 models. Additionally, the desktop environment serves as an initial baseline to compare against the edge hardware. The desktop computer used in this praxis is equipped with an AMD Ryzen 7700x CPU, 64 GB of DDR5 memory, and an NVIDIA RTX 5060TI with 16 GB of video memory. These specifications are more than adequate to train and test the YOLOv8 models used in this praxis. Additionally, the desktop computer uses several pieces of software. This includes Python 3.10, Miniconda, and Python libraries like Ultralytics, PyTorch, Tensorflow, NumPy, Pandas, and Jupyter. The desktop environment also provides an adequate environment to host the virtual environment using Unreal Engine 4.27 to simulate real-world lighting conditions and visual scenarios.

### 3.6.2 Nvidia Jetson Orin Nano Test Environment

A Raspberry Pi 5 was initially selected to conduct model testing. However, after several performance limitations were encountered, an NVIDIA Jetson Orin Nano Super Developer Kit was selected for this praxis. This upgrade in hardware provides significant

increases in computational power as well as better performance with performance with artificial intelligence models. There are several implementations of previous versions of NVIDIA's edge device ready hardware that were incorporated across several academic papers that referenced during the research phase of this praxis that provide insight into the potential benefits of leveraging similar hardware for use on small unmanned aerial vehicles.

The NVIDIA Jetson Orin Nano Super Developer Kit (Figure 2) is equipped with a 6-core ARM Cortex-A78AE 64 bit-processor, an NVIDIA GPU with 1024 CUDA cores and 32 Tensor cores, 8GB of LPDDR5, and 256GB of onboard storage (NVIDIA, 2024). These specifications provide a significant upgrade in hardware and performance in comparison to the previously tested Raspberry Pi 5 and additionally provides 67 tera operations per second (TOPS) when supporting artificial intelligence operations (NVIDIA, 2024).

The addition of this device into the prototyping environment requires configuration of the device and onboard software. The NVIDIA Jetson Orin Nano leverages the preconfigured JetPack software development kit (SDK) and includes Ubuntu 20.04 LTS, the CUDA toolkit, cuDNN, TensorRT, OpenCV framework, and the NVIDIA container runtime. Following the initial configuration of the hardware and operating system, the packages for JupyterLab, Python, Python libraries, PyTorch, and YOLOv8 are installed on the Jetson Orin Nano.

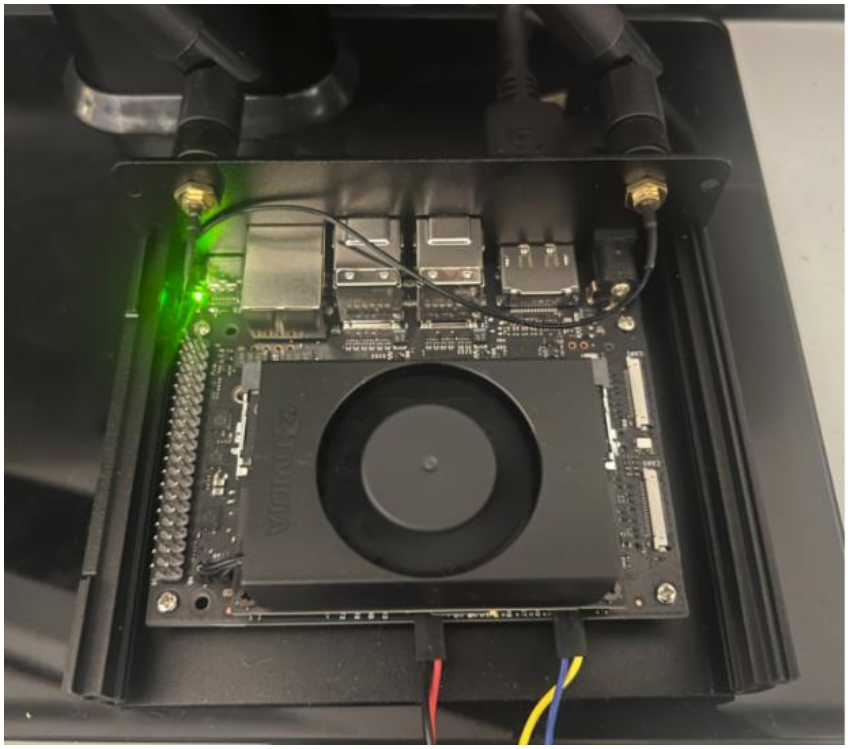

**Figure 2: NVIDIA Jetson Orin Nano Super Developer Kit**

### 3.7 Simulated Environment

Several academic articles referenced during the research phase of the praxis identify various software programs including Microsoft AirSim which leverages the Unreal Engine to provide photorealistic and dimensionally accurate simulated environments to test various artificial intelligence models. Because of this, the Unreal Engine v4.27 is leveraged to provide a photorealistic environment necessary to test the impact of various lighting conditions on the YOLOv8 model's real-time performance while deployed on an edge device.

Several referenced articles highlight the need for a real-time test environment to evaluate the efficacy of an optimized deep learning computer vision model on UAV platforms. To accomplish this goal, Off World Interactive's Squad Software Development Kit (SDK) running on the Unreal Engine v4.27 (Figure 3) is leveraged to

provide a suitable platform to conduct test and evaluation of vision models trained on the consolidated Russian military datasets. This simulated environment provides an adequate platform with high-fidelity imagery that can replicate various visual environments which could emulate real-world conditions. Additionally, this simulated environment provides a visually accurate test environment, with a high degree of control over environmental variables to further enhance the repeatability and consistency of the experimentation.

**Figure 3: Squad SDK Running in Unreal Engine v4.27**

### 3.8 Equations

This praxis uses several equations to measure the performance of a deep learning model. This includes metrics like the confusion matrix, precision, recall, f1-score, intersection over union, and mean average precision equations.

#### 3.8.1 Bounding Box

The bounding box is an equation that describes the location of an object within an image (Redmon et al., 2016). The location information identified by this equation can be used with object detection models like YOLO to identify and predict object locations in images (Redmon et al., 2016). An example of the bounding box equation is:

$$Bounding\ Box\ =\ (x, y, w, h)$$

**Equation 1: Bounding Box**

(Redmon et al., 2016)

**Figure 4: Bounding Box Example**

#### 3.8.2 Confusion Matrix

The confusion matrix is a metric that describes the relationship between true positive, true negative, false positive, and false negative predictions (Dou et al., 2025). This metric is used to calculate additional equations used in this praxis. An example of the confusion matrix is:

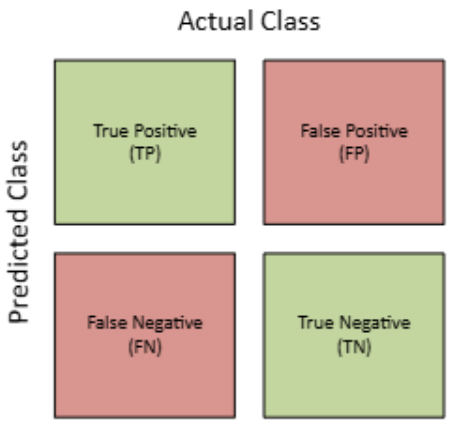


**Figure 5: Confusion Matrix Example**

(Dou et al., 2025)

### 3.8.3 Precision

Precision is a metric that describes the ratio of true positive predictions in comparison to all positive predictions (Sanikommu et al., 2025). The formula for precision is:

$$precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

**Equation 2: Precision**

(Sanikommu et al., 2025)

### 3.8.4 Recall

The recall metric is an equation that describes the proportion of true positive predictions against all actual positive predictions that a model observes (Sanikommu et al., 2025). The formula for recall is:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

**Equation 3: Recall**

(Sanikommu et al., 2025)

### 3.8.5 F1-Score

The f1-score is a metric that combines the mean of precision and recall providing a balanced metric of the two equations (Sanikommu et al., 2025). The formula for f1-score is:

$$F1\ Score\ =\ 2\ \cdot\ \frac{precision\ \cdot\ recall}{precision\ +\ recall}$$

**Equation 4: F1 Score**

(Sanikommu et al., 2025)

### 3.8.6 Intersection over Union (IoU)

The Intersection over Union (IoU) metric describes how correct a model's bounding box location is when compared to the actual bounding box location on an image (Sanikommu et al., 2025). The formula for Intersection over Union is:

$$\boldsymbol{IoU} = \frac{True\ Positives}{(True\ Positives + False\ Positives + False\ Negatives)}$$

**Equation 5: Intersection over Union (IoU)**

(Sanikommu et al., 2025)

### 3.8.7 Mean Average Precision

The Mean Average Precision (mAP) metric describes the average precision of the positive predictions across several classes (Qu et al., 2025). The formula for Mean Average Precision is:

$$mAP\ =\ \frac{1}{k}\sum_{i=1}^{k} AP$$

**Equation 6: Mean Average Precision (mAP)**

(Qu et al., 2025)

The Mean Average Precision 50 (mAP50) metric is a variant of the mAP equation. This equation additionally describes the average precision of true positive predictions when the IoU is at fifty percent (De Oliveira Barbosa et al., 2025). The formula for mAP50 is:

$$mAP_{50} = \left(\frac{1}{N}\right) \cdot \sum [AP < sub > c, IOU = 0.50 < sub >] \; from \, c = 1 \; to \; N$$

**Equation 7: Mean Average Precision 50 (mAP50)**

(De Oliveira Barbosa et al., 2025)

The Mean Average Precision 50-95 (mAP50-95) metric is another variant of the mAP equation and describes the true positive predictions only with an IoU threshold between 50% and 95% (De Oliveira Barbosa et al., 2025). The formula for mAP50-95 is:

$$mAP_{95} = \left(\frac{1}{N}\right) \cdot \sum [AP < sub > c, IOU = 0.95 < sub >] \; from \, c = 1 \; to \; N$$

**Equation 8: Mean Average Precision 50-95 (mAP50-95)**

(De Oliveira Barbosa et al., 2025)

# Chapter 4—Results

## 4.1 Introduction

This chapter details the application of the methodology to pre-process and combine several datasets, train a YOLOv8m model, and administer several experiments leveraging an NVIDIA Jetson Orin Nano Super Developer Kit to test the efficacy of deploying a deep learning CNN for use on edge devices with limited computational resources. The experiments in this chapter provide insight into the application of a deep learning CNN across several environments that simulate common constraints in the deployment on UAVs to include power constraints, limited memory, and operation in low-light environments.

### 4.1.1 Limitations and Constraints

There are several limitations experienced during the configuration and deployment of the test environments. These range from limitations identified during the initial configuration and setup to limitations identified in the testing and collection of visual data in the simulated environment. Inconsistencies across all three datasets used in this praxis require additional steps to align labels and bounding box data. Additionally, poor visual image data within the *Oryx 2022 Russia Ukraine War* dataset require additional cleaning and pre-processing to mitigate ingestion of poor visual data into the model. Imbalance of label classes are another limitation due to the limited amount of consolidated Russian vehicle data available for this praxis. Because of this, several mitigation steps are implemented to reduce the impact of the imbalance.

### 4.2 Dataset

The three image datasets of Russian military vehicles share similar but not identical label methodologies, some opting for more specific model labels and others favoring the overall vehicle class. To mitigate the discrepancy between dataset label methodologies, the use of an overarching military vehicle class is utilized to organize all the common armored vehicle models. All the different vehicle models are grouped into the primary vehicle categories with labels of Main Battle Tank (MBT), Infantry Fighting Vehicle (IFV), Armored Personnel Carrier (APC), Light Armored Car (LAV_CAR), and Light Armored Truck (LAV_TRUCK). The selection of this label naming convention provides organization of different military vehicle models into classes that share similar visual characteristics to approximately 9,000 images across five classes.

### 4.3 Data Cleaning

Because of the inclusion of heavily damaged and unrecognizable images of vehicles (Figure 6) alongside easily recognizable images (Figure 7), the *Oryx 2022 Russia Ukraine War* dataset requires additional cleaning before it can be included in the model training. To clean this data, several methods are used including pre-processing of the data with a YOLOv8x model to generate bounding box locations as well as manual imputation of unrecognizable images of vehicles. In addition to these techniques, methods like oversampling and image augmentations are also used in this praxis to assist during model training.

**Figure 6: Oryx 2022 Russia Ukraine War Dataset, Example of Destroyed Vehicles**

**Figure 7: Oryx 2022 Russia Ukraine War Dataset, Example of Intact Vehicles**

### 4.3.1 Optimization, Data Imbalance

The three datasets are combined into a single consolidated set of images and labels. After this, the consolidated dataset is analyzed for class imbalance and label consistency. The dataset is divided into 70% training, 20% validation, and 10% test splits

to ensure adequate image data for training while maintaining separate splits to validate model performance on unseen images. The consolidated dataset initially consists of 5,601 usable images of Russian military vehicles (Table 1). However, the distribution of images is not even across all five classes following consolidation. In particular, the IFV and MBT classes are the most prevalent images in the dataset with more than 2,000 images per class while the APC, LAV_CAR, and LAV_TRUCK have substantially fewer images with an average of 500 images per class. To mitigate underrepresentation of APC, LAV_CAR, and LAV_TRUCK classes during the model training, oversampling is implemented only to the training split to increase the underrepresented image classes and ensuring isolation of images between splits. Oversampling the underrepresented image classes increases the total consolidated dataset to 8,939 images: 7,258 training images, 1,120 validation images, and 561 test images (Table 2). In addition to addressing imbalanced classes prior to model training, several optimization steps are also implemented including normalization, image resizing to 640x640 pixels, as well as several augmentation techniques including random crops, image flipping, image rotation, and image distortion as identified in the literature review.

**Table 1: Dataset Pre-Class Balance**

| **Pre-Class Balance.**<br>**Total Images:** 5601 | |
|---|---|
| **Class Labels:** | **Quantity:** |
| **IFV** | 2229 |
| **APC** | 435 |
| **MBT** | 2715 |
| **LAV_CAR** | 429 |
| **LAV_TRUCK** | 630 |

**Table 2: Dataset Post-Class Balance**

| **Post-Class Balance.**<br>**Total Images:** 8939 | | |
|---|---|---|
| **Training Split Labels:** | **Class Name:** | **Quantity:** |
| | IFV | 1560 |
| | APC | 1788 |
| | MBT | 1923 |
| | LAV_CAR | 1824 |
| | LAV_TRUCK | 1299 |
| Total Training Images: | 7258 | |
| **Validation Split Labels:** | **Class Name:** | **Quantity:** |
| | IFV | 441 |
| | APC | 103 |
| | MBT | 527 |
| | LAV_CAR | 74 |
| | LAV_TRUCK | 140 |
| Total Validation Images: | 1120 | |
| **Test Split Labels:** | **Class Name:** | **Quantity:** |
| | IFV | 228 |
| | APC | 34 |
| | MBT | 265 |
| | LAV_CAR | 51 |
| | LAV_TRUCK | 57 |
| Total Test Images: | 561 | |

### 4.4 Model Training

Several iterations of model training are conducted in this praxis. First, an initial model is developed using the *WarTech_2.0* image dataset to serve as an initial method to clean and apply bounding box data to images within the *Oryx 2022 Russia Ukraine War* dataset. Second, after consolidation and evaluation of the consolidated dataset comprised

of *WarTech_2.0, Oryx 2022 Russia Ukraine War,* and *Military_Vehicles* datasets, an initial YOLOv8x model is trained on the desktop environment to serve as an initial baseline point of evaluation of the model's performance without resource constraints prior to the training of a YOLOv8m model for export to the NVIDIA Jetson Orin Nano Super Developer Kit.

The initial model consists of a YOLOv8x (extra-large size) model comprised of approximately 68 million parameters run for 25 epochs to determine the effectiveness of the hyperparameter optimizations prior to the training of the final smaller model that consisted of a YOLOv8m (medium size) model comprised of approximately 26 million parameters. After the YOLOv8x model demonstrated adequate performance (Figure 8), the YOLOv8m model is trained for 100 epochs with hyperparameters (Table 3) set to assist in the model's training for small object detection as well as data augmentation to mitigate the impacts of imbalanced classes. The results (Figure 9) of the YOLOv8m model training and test split validation demonstrate a strong overall ability to accurately detect various military vehicle classes while maintaining real-time inference speeds (Table 4).

**Table 3: YOLOv8x and YOLOv8m Training Hyperparameters**

| YOLOv8 Training Hyperparameters: | | | |
|---|---|---|---|
| **YOLOv8x (extra-large):** | | **YOLOv8m (medium):** | |
| **Epochs:** | 25 | **Epochs:** | 100 |
| **Imgsz:** | 640x640 | **Imgsz:** | 640x640 |
| **Batch:** | 8 | **Batch:** | 8 |
| **Device:** | 0 | **Device:** | 0 |
| **Seed:** | 42 | **Seed:** | 42 |
| **Workers:** | 8 | **Workers:** | 8 |
| **Val:** | True | **Val:** | True |
| **Verbose:** | True | **Verbose:** | True |
| **Augment:** | True | **Augment:** | True |
| **Close_Mosaic:** | 5 Epochs | **Close_Mosaic:** | 20 Epochs |
| **Patience:** | 10 | **Patience:** | 25 |

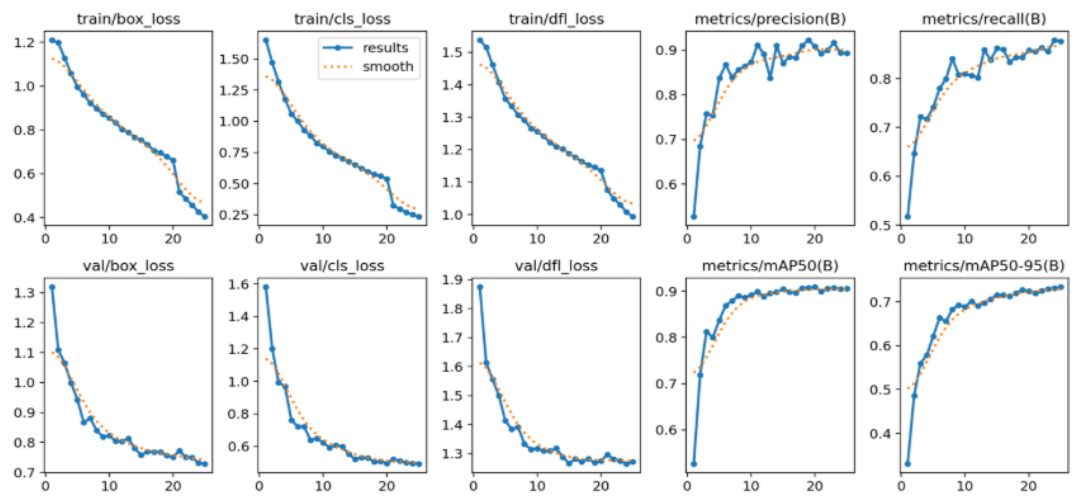


**Figure 8: YOLOv8x (extra-large) Training Results (25 epochs)**

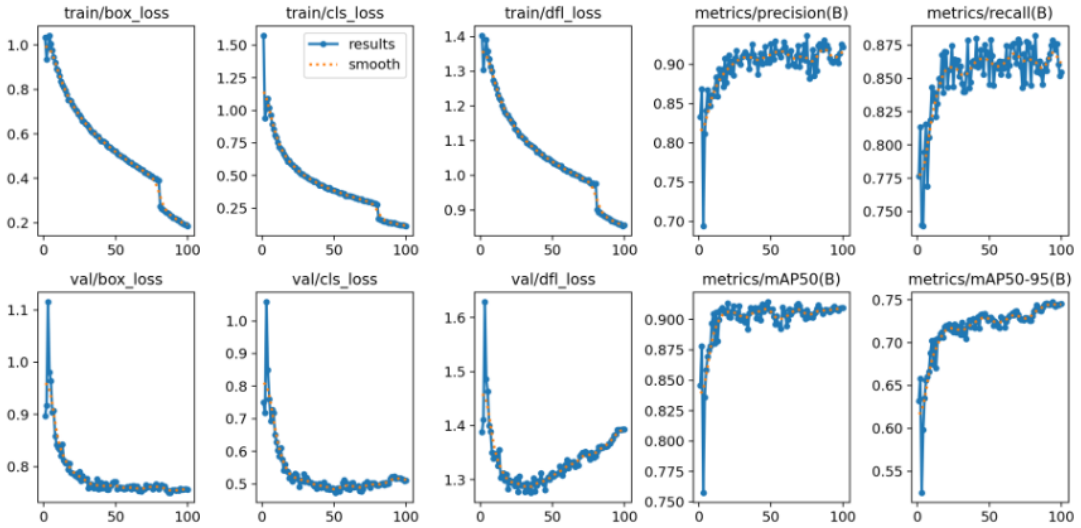


**Figure 9: YOLOv8m (medium) Training Results (100 epochs)**

**Table 4: YOLOv8m Model Results**

| YOLOv8m (medium) Training Results: 100 epochs | |
|---|---|
| **Precision:** | 0.906 |
| **Recall:** | 0.874 |
| **mAP50:** | 0.919 |
| **mAP50-95:** | 0.758 |
| **Inference Speed:** | 127 FPS (Desktop) |
| **YOLOv8m (medium) Test Split Results** | |
| **Precision:** | 0.921 |
| **Recall:** | 0.859 |
| **mAP50:** | 0.911 |
| **mAP50-95:** | 0.743 |
| **Inference Speed:** | 149 FPS (Desktop) |

### 4.5 Deployment on Edge Computer

To deploy the YOLOv8m model to the NVIDIA Jetson Orin Nano Super Developer Kit, the model is first trained on the desktop environment and then exported to the NVIDIA Jetson Orin Nano where it is run in Python within a container. To establish a baseline for conducting tests for the research questions on the NVIDIA Jetson Orin Nano, the test split image data is used to provide unseen image data for the model (Figure 10). The results of the baseline test on the NVIDIA Jetson Orin Nano demonstrate consistent object detection performance in-line with the desktop baseline tests; however, with lower inference speeds on the edge device (Table 5).

**Figure 10: YOLOv8m Test Split Predictions Sample**

The NVIDIA Jetson Orin Nano platform provides various capabilities for testing methodologies to validate the efficacy of an AI model for deployment on edge devices. These include the ability to manually control the power across several power modes that could be used to limit total power consumption and processing speeds as well as restrict system memory capacity to simulate performance on resource constrained systems.

**Table 5: YOLOv8m Individual Class Results**

| YOLOv8m Test Split Results on NVIDIA Jetson Orin Nano | | | | | |
|---|---|---|---|---|---|
| **Class:** | **Precision** | **Recall** | **mAP50** | **mAP50-95** | **Inference** |
| **IFV** | 0.865 | 0.909 | 0.907 | 0.754 | 25 FPS |
| **APC** | 0.935 | 0.851 | 0.885 | 0.728 | 25 FPS |
| **MBT** | 0.913 | 0.909 | 0.964 | 0.788 | 25 FPS |
| **LAV_CAR** | 0.926 | 0.765 | 0.859 | 0.697 | 25 FPS |
| **LAV_TRUCK** | 0.956 | 0.877 | 0.93 | 0.737 | 25 FPS |
| **Overall:** | 0.919 | 0.86 | 0.909 | 0.741 | 25 FPS |

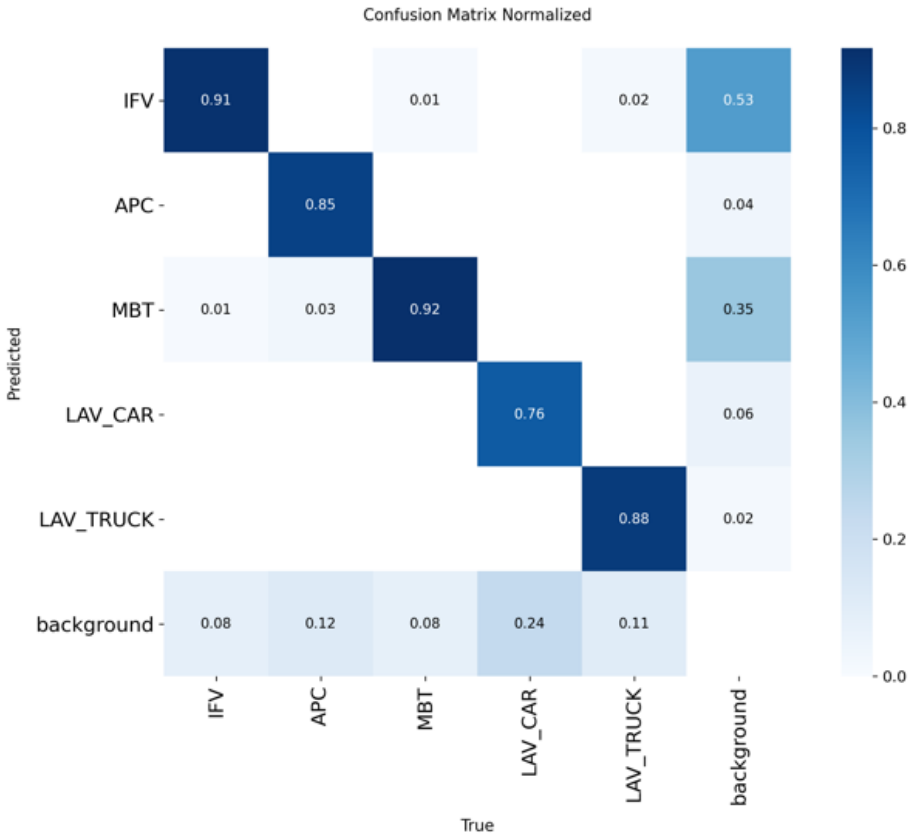


**Figure 11: YOLOv8m Test Split Results, Confusion Matrix**

### 4.6 Object Detection in Simulated Environment

To test the models' capabilities on edge devices, it is necessary to establish an environment that may provide control over environmental variables to ensure adequate testing of various parameters as well as reproducibility. There are several considerations necessary to configure the simulated environments for testing the model's capabilities to include consistency and repeatability of the test environments, control over several environmental variables such as lighting and weather, as well as latency and throughput from the test environment to the edge device. To achieve this, several steps are taken to set up and standardize the simulation environment. The simulated environment is standardized across all tests and consists of a single environment with photorealistic and dimensionally accurate environmental details as well as several Russian military vehicles

from the primary vehicle classes to include tracked infantry fighting vehicles (IFV), wheeled armored personnel carriers (APC), main battle tanks (MBT), light armored cars (LAV_CAR), and light armored trucks (LAV_TRUCK) (Figure 12).

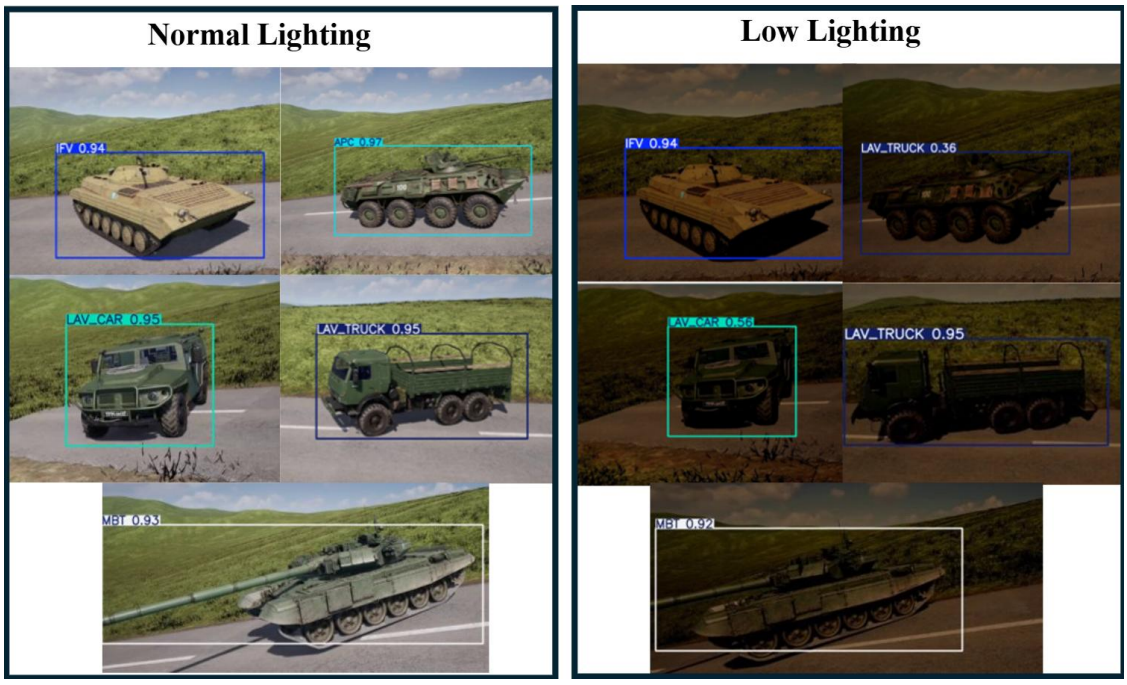


**Figure 12: Unreal Engine v4.27, Lighting Conditions Tests**

The implementation of a simulated environment using the Squad SDK in Unreal Engine v4.27 provides a controlled method to test against environmental lighting conditions in leu of access to a real-world test environment. Ideal overhead light conditions and low-light conditions are used within the Unreal Engine simulated environment to compare the effectiveness of the model deployed on an edge device. Both lighting conditions are applied to the same simulated environment consisting of the same layout of photorealistic assets and camera movement to provide a consistent mechanism to test the efficacy of the deep learning model running on the NVIDIA Jetson Orin Nano Super Developer Kit against simulated Russian military assets. Two 40 second videos recorded at 30 FPS are captured in the Unreal Engine simulated environment under

normal and low light conditions resulting in the generation of simulated test images consisting of approximately 1200 frames per video.

### 4.7 Hypothesis and Research Question Results

The development of the test methodologies and test environments provides a path to conduct test and evaluation of the YOLOv8m model's performance to answer several research questions in this praxis. The first research question examines the impact on real-time object detection accuracy in low-light conditions when using Convolutional Neural Networks (CNNs) on edge devices. This first area of research introduces several variables into the test and evaluation of the model within the simulated environment. Factors such as environment and object lighting, camera settings, and camera resolution are variables that are reproduced within the simulated environment and considered during the test and evaluation. The second research question examines how the reduced computational resources of edge devices affect the inference time and accuracy for object detection using lightweight CNNs such as YOLOv8m. This research question is heavily dependent on hardware and model selection and depends on factors related to the number of CPU cores, number of GPU cores, power consumption, and clock speed, and its impact on model inference times. The third research question examines how the limited memory capacity on edge devices influences the model's inference speed and object detection accuracy and if reduction to the inference resolution may improve real-time inference performance without substantial degradation to the model's precision. This research question is dependent on several factors including hardware selection, model selection, as well as model training and deployment on an edge device and is tested

through the monitoring of the edge device's memory resources when implementing the model.

### 4.7.1 Research Question 1

An adequate amount of data was collected to test the efficacy of the YOLOV8m deep learning model across various visual conditions within a simulated environment and answer the research question of *"what would be the impact to real-time object detection accuracy in low-light conditions using Convolutional Neural Networks (CNNs) on edge devices?"*. There are several considerations that factor into the testing of the deep learning model across the different lighting scenarios in the simulated environment. These include two lighting scenarios ranging from ideal overhead light conditions similar to the sun at mid-day and low-light conditions similar to evening sunlight. Across both lighting scenarios, the same flight path is simulated and recorded to ensure repeatability across tests. Several 3D objects of Russian military vehicles from each class are placed throughout the flight path to incorporate different angles and perspectives within a rural environment.

The results of the tests demonstrate the impact that the reduction in overall lighting conditions has on the deep learning model's ability to successfully detect different classes of Russian military vehicles with a high degree of confidence at real-time inference speeds. In the normal light conditions test, the YOLOv8m model's performance demonstrates a strong overall ability to detect the Russian military assets across all the vehicle classes with real-time inference speed running on the NVIDIA Jetson Orin Nano Super Developer Kit.

**Table 6: Research Question 1, Normal Light Conditions**

| Unreal Engine v4.27: Normal Light Condition Test Results | | | | | |
|---|---|---|---|---|---|
| **Class:** | **Precision** | **Recall** | **mAP50** | **mAP50-95** | **Inference** |
| **IFV** | 1.00 | 0.975 | 0.995 | 0.995 | 33 FPS |
| **APC** | 0.998 | 1.00 | 0.995 | 0.995 | 33 FPS |
| **MBT** | 0.997 | 1.00 | 0.995 | 0.994 | 33 FPS |
| **LAV_CAR** | 0.996 | 1.00 | 0.995 | 0.995 | 33 FPS |
| **LAV_TRUCK** | 0.996 | 0.992 | 0.995 | 0.995 | 33 FPS |
| **Overall:** | 0.998 | 0.993 | 0.995 | 0.995 | 33 FPS |

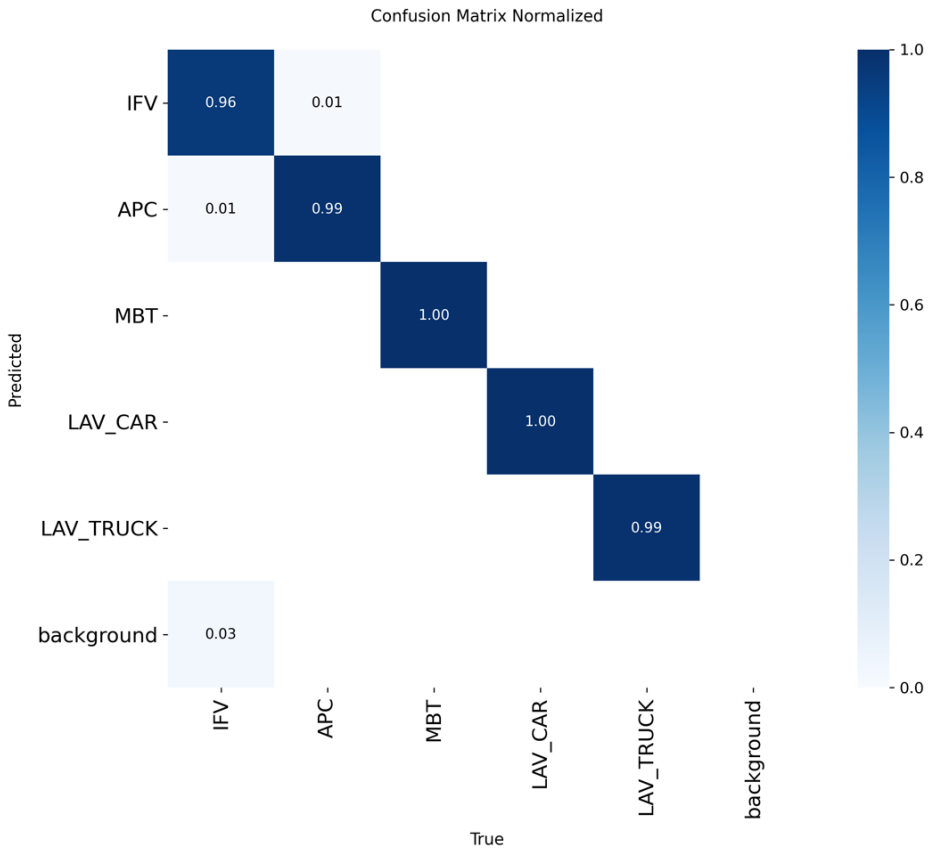


**Figure 13: YOLOv8m Model Confusion Matrix for Normal Lighting Test**

The result of the normal light conditions test demonstrates a strong overall ability of the model to generalize to the simulated environment across all vehicle classes with

very limited missed detections. The normal light conditions and non-complex background provide the model with ideal detection conditions within the simulated environment. These results serve as a baseline for comparison of low-light conditions in the subsequent test.

The result of the low-light conditions test demonstrates a reduction in overall performance of the YOLOv8m model in comparison to the normal light conditions test. In particular, the precision, recall, mAP50, and mAP50-95 are significantly lower overall in comparison to the normal light conditions test results and demonstrate an increase in missed detections across classes. The detection performance for the IFV class demonstrates the least impact by the decreased lighting conditions with a 5.2% reduction in precision, a 1.1% reduction in recall, a 2.3% reduction in mAP50, and a 4.8% reduction in mAP50-95. The IFV class demonstrates the strongest performance out of all classes under simulated low-light conditions. Next, the detection performance of the APC class demonstrates a precision of 1.0 with a substantially diminished recall of 0.142 demonstrating an 85.8% reduction in the YOLOv8m model's vehicle detection of the APC class. Furthermore, the APC class demonstrates a reduced mAP50 of 0.765 and a mAP50-95 of 0.684, a 23.1% and 31.2% reduction in overall precision, respectively. Next, the detection performance of the MBT class demonstrates a 20.7% reduction in precision and a 4.5% reduction in recall. Furthermore, the MBT class exhibits a 1.6% reduction in mAP50 and a 24.5% reduction in mAP50-95 performance. Next, the LAV_CAR class maintains a high degree of overall precision with an 8.5% drop in mAP50 performance but demonstrates a 44.3% reduction in recall performance to 0.557. Last, the LAV_TRUCK class demonstrates a 26.6% reduction in mAP50. Although the

LAV_TRUCK class maintained a high degree of recall performance, it demonstrates a reduction in correct identification of the LAV_TRUCK class. The results of testing the YOLOV8m model under simulated low-light conditions demonstrate variances in performance across vehicle classes. While IFV and MBT classes maintain strong performance under low-light conditions, the APC, LAV_CAR, and LAV_TRUCK classes demonstrate varying levels of diminished performance.

**Table 7: Research Question 1, Low-Light Conditions**

| **Unreal Engine v4.27: Low-Light Condition Test Results** | | | | | |
|---|---|---|---|---|---|
| **Class:** | **Precision** | **Recall** | **mAP50** | **mAP50-95** | **Inference** |
| **IFV** | 0.948 | 0.964 | 0.972 | 0.947 | 33 FPS |
| **APC** | 1.00 | 0.142 | 0.765 | 0.684 | 33 FPS |
| **MBT** | 0.79 | 0.955 | 0.979 | 0.75 | 33 FPS |
| **LAV_CAR** | 1.00 | 0.557 | 0.91 | 0.831 | 33 FPS |
| **LAV_TRUCK** | 0.37 | 1.00 | 0.73 | 0.688 | 33 FPS |
| **Overall:** | 0.822 | 0.724 | 0.871 | 0.78 | 33 FPS |

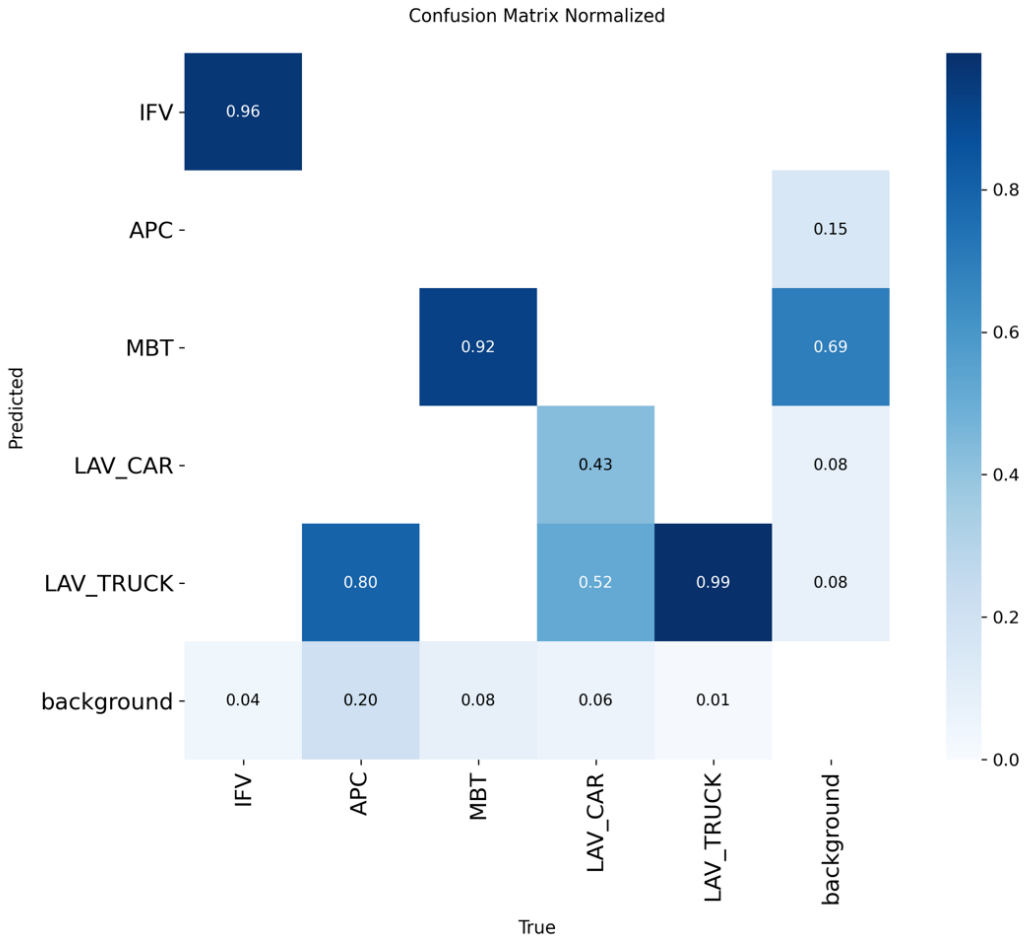


**Figure 14: YOLOv8m Model Confusion Matrix for Low-Lighting Test**

### 4.7.2 Hypothesis 1

The hypothesis of *"A CNN (like YOLO) can maintain at least 80% detection accuracy in low-light conditions on edge devices"* is partially correct. The results of the various lighting tests within the simulated environment demonstrate that the YOLOv8m model maintained an 80% overall detection accuracy (mAP50) of adversary military assets in low-light conditions. However, this hypothesis was not correct for all classes individually as several classes failed to maintain at least an 80% mAP50 score relative to the model's detection performance under normal lighting conditions. The YOLOv8m model's performance in low-light conditions demonstrates varying levels of reductions in

performance across classes. The IFV class maintains strong overall detection metrics in low-light conditions with only 2.3% reduction in mAP50 and maintains strong recall at 0.964 indicating an ability to continue to detect the presence of objects even under low-light conditions. The MBT class demonstrates a similar reduction in performance with a 1.6% reduction in mAP50 while maintaining strong recall, only dropping by 4.5% in low-light conditions. The LAV_CAR class demonstrates some degradation to performance with an 8.5% reduction in mAP50 and a 44.3% reduction in recall performance indicating the YOLOv8m model's difficulty in seeing the vehicle class in reduced lighting conditions. The APC class demonstrates diminished performance with a reduction of mAP50 by 23.1%. Additionally, the APC class has substantially diminished recall performance, demonstrating a reduction by 85.8%. The LAV_TRUCK class demonstrates the lowest performance of all classes with a 26.6% reduction in mAP50. The results of this experiment demonstrate the impact that low-light conditions may have on the performance of the YOLOv8m model. In addition to lowered mAP50 performance across all classes, recall performance was also degraded. Although the APC and LAV_CAR classes continue to demonstrate high precision under low-light conditions, both classes struggle with 85.6% and 44.3% reductions in recall, respectively due to a diminished ability to detect the presence of objects. Overall, the YOLOv8m model maintained a mAP50 of 0.871 under simulated low-light conditions, a reduction of 12.4% from the normal light conditions baseline results. Because of this, Hypothesis 1: *A CNN (like YOLO) can maintain at least 80% detection accuracy in low-light conditions on edge devices,* is correct relative to the overall performance of the model across the dataset but is not correct for all classes individually.

### 4.7.3 Research Question 2

Adequate data was collected while testing the efficacy of the YOLOv8m deep learning model on the NVIDIA Jetson Orin Nano Super Developer Kit, to answer the research question of *"how do reduced computational resources affect inference time and object detection accuracy for CNNs on edge devices?"*.

To test the impact of reduced computational resources on an edge device, the NVIDIA Jetson Orin Nano Super Developer Kit is configured with three power settings for each test. This configuration is simply implemented through a built-in capability of the Jetson Orin Nano's software that enables selection of various power modes to restrict the device to use only 7W, 15W, and 25W of power (Figure 15). All other variables in this test remain constant to isolate the impact of power consumption limitations on the performance of the deep learning model.

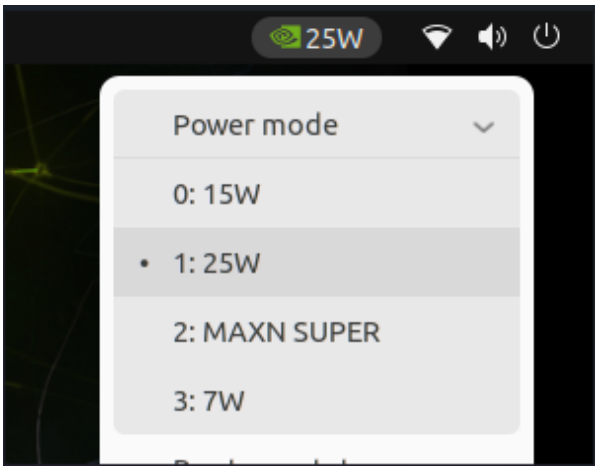


**Figure 15: NVIDIA Jetson Orin Nano Super Developer Kit Power Modes**

Three tests are conducted for this second research question and leverage the test split images when running on the Jetson Orin Nano platform. The first test with 25W of power establishes a base level of performance on the edge device and demonstrates

strong precision, recall, and mAP scores which remain consistent across all three tests. The primary performance metric to examine the impact of power level is the overall inference speed of the model. The first test with 25W demonstrates an inference speed of 25.5 FPS, adequate for real-time detection. The second test with 15W demonstrates a drop in inference speed to 18.6 FPS. The third test with 7W demonstrates a more substantial drop in inference speed to 8.2 FPS (Table 8).

**Table 8: Research Question 2 Results**

| Restricted Power Impacts to Inference Speed Results: | | | | | |
|---|---|---|---|---|---|
| | **Precision** | **Recall** | **mAP50** | **mAP50-95** | **Inference Speed** |
| **Test 1: 25W (baseline)** | 0.919 | 0.86 | 0.909 | 0.741 | 25.5 FPS |
| **Test 2: 15W** | 0.919 | 0.86 | 0.909 | 0.741 | 18.6 FPS |
| **Test 3: 7W** | 0.919 | 0.86 | 0.909 | 0.741 | 8.2 FPS |

### 4.7.4 Hypothesis 2

The hypothesis of *"Reduced computational resources increase inference time by 40%, leading to delayed object detection but < 10% decrease in accuracy"* is partially correct. The results of the reduction in computational resources demonstrate an approximate decrease in inference speed of 27% from 25W to 15W. This decrease did not impact the model's detection accuracy. The results of the decrease in power from 25W to 7W demonstrate a substantial decrease in inference speed of 67% without substantial impact on the model's detection accuracy. Additionally, the results of the decrease in power from 15W to 7W demonstrate a decrease of 55%. The results of the 7W power tests demonstrate a substantial reduction in real-time detection capability of the model. Because of this, Hypothesis 2 was correct relative to the reduction of power

from 25W to 15W as it had less than a 40% increase in inference time while maintaining its detection accuracy, confirming the hypothesis. However, Hypothesis 2 was incorrect relative to the reduction of power from 25W to 7W and 15W to 7W as it did cause a significant reduction in inference speed greater than 40% and without substantial impact to the model's detection accuracy.

### 4.7.5 Research Question 3

An adequate amount of data was collected to answer the research question of "*How does limited memory capacity on edge devices influence the model inference speed and object detection accuracy of CNNs?"* when testing on the YOLOv8m model on the NVIDIA Jetson Orin Nano. To test impacts on the reduction of system memory on the inference speed and object detection accuracy of the YOLOv8m model, memory limits are set to 8GB and 4GB to identify the impact of the reduced memory. Within the Linux environment on the Jetson Orin Nano, use of "stress-ng –vm1 –vm-bytes <*memory amount*> --timeout <*time in seconds*>" command provides a method to restrict available system memory for the container to conduct each test.

The baseline test with 8GB of system memory with inference resolution set to 640x640 established a baseline level of performance when run against the test split image data on the Jetson Orin Nano (Table 9). These initial results demonstrate strong overall performance and real-time inference speed of 25.5FPS. The second test consists of the reduction of the system memory to 4GB while maintaining an inference resolution of 640x640. The results of this test demonstrate a 27% reduction in inference speed of the model on the Jetson Orin Nano, reducing it from 25.5 FPS to 18.4 FPS. The third test consists of the reduction of available system memory to 4GB and the reduction of

inference resolution by 50% to 320x320. The results of this test demonstrate a 1% reduction in precision, a 6% reduction in recall, a 3.6% reduction in mAP50, a 4.4% reduction in mAP50-95, and a 207% increase in inference speed. A fourth and fifth test further reducing available system memory to 2GB with inference resolutions of 640x640 and 320x320 both result in an insufficient amount of system resources and failure to complete the tests.

**Table 9: Research Question 3 Results**

| **Memory and Resolution Impacts to Inference Speed Results:** | | | | | |
|---|---|---|---|---|---|
| | **Precision** | **Recall** | **mAP50** | **mAP50-95** | **Inference Speed** |
| **Test 1: 8GB @ 640x640** | 0.919 | 0.86 | 0.909 | 0.741 | 25.5 FPS |
| **Test 2: 4GB @ 640x640** | 0.919 | 0.86 | 0.909 | 0.741 | 18.4 FPS |
| **Test 3: 4GB @ 320x320** | 0.918 | 0.807 | 0.876 | 0.708 | 56.5 FPS |
| **Test 4: 2GB @ 640x640** | Failed to run due to insufficient resources. | | | | |
| **Test 5: 2GB @ 320x320** | Failed to run due to insufficient resources. | | | | |

### 4.7.6 Hypothesis 3

The hypothesis of *"Reducing image resolution by 50% improves inference speed by 20% for memory capacity-limited edge devices without reducing object detection accuracy by more than 10%*" is correct. The results of reductions in available system memory in combination with reduction in inference resolution demonstrate the impact of limited memory resources on the real-time performance of the YOLOv8m model. The

reduction to 4GB of system memory yields a decrease of 27% in inference speed while maintaining an inference resolution of 640x640. The reduction to 4GB of system memory with a reduction in inference resolution by 50% to 320x320 yields a slight decrease in performance accuracy ranging from 1-6% across evaluated metrics and increases the inference speed by 207%. Further reduction in system memory beyond this yields an insufficient amount of system resources for the model to successfully complete a test.

Because of this, Hypothesis 3 is correct. Comparison of the results of the second test with 640x640 inference resolution and the third test with 320x320 inference resolution with reduced system memory of 4GB, result an overall detection accuracy decrease by 1-6% across the evaluated metrics while yielding a 207% increase in inference speed confirming the hypothesis. However, further reduction of the system memory to 2GB results in overall failure of the model due to complete resources necessary to complete inference.

# Chapter 5—Discussion and Conclusions

## 5.1 Discussion

The proliferation of small, unmanned aircraft systems (sUAS) in military applications has a significant impact on modern military strategy (Gerstein & Leidy, 2024). The rapid scalability and ease of access to this technology have increased its prevalence on the modern battlefield for various tactical capabilities to include intelligence, surveillance, and reconnaissance (ISR) and kinetic attacks when equipped with munitions (Pledger, 2021). Additionally, the implementation of electronic warfare (EW) countermeasures to defeat traditionally wirelessly controlled sUAS devices is noted to have a major impact on the success rate and deployment of these devices in contested military environments in recent history (Watling & Reynolds, 2023). In line with the goals of the DOD Replicator Initiative, this praxis details the implementation of a convolutional neural network (CNN) using the YOLOv8m architecture on an NVIDIA Jetson Orin Nano Super Developer Kit. This implementation is used to test the potential deployment of deep learning for use on small, unmanned aircraft systems (sUAS) for real-time detection of Russian military assets in disrupted environments. Additionally, this praxis includes several experiments to test the efficacy of a lightweight detection model on an edge device under different environmental conditions. The results of the experiments in this praxis demonstrate the potential feasibility of leveraging a YOLOv8m model for real-time detection of adversary military assets from an edge device.

## 5.2 Conclusions

This praxis led to the implementation of a real-time object detection model for use on edge devices built on the YOLOv8m architecture. The deployment of this deep learning model on an NVIDIA Jetson Orin Nano demonstrates the potential application of artificial intelligence models to detect adversary military assets from edge devices like sUAS in real-time in EW disrupted environments.

Chapter 1 of this praxis identifies the significant role that sUAS has in national security and military contexts. The chapter includes the role that EW has in countering sUAS. In addition to referencing background information from a variety of sources, the chapter includes the thesis of this praxis: *A Convolutional Neural Network tailored for edge devices will enhance drone resilience to electronic warfare by over 90%, enabling detection and identification of adversarial military assets in electronic warfare-contested environments (Molloy, 2024; Sayler, 2024; Watling & Reynolds, 2023).*

Chapter 2 asses a collection of reference materials used in this praxis. The chapter includes subsections outlining crucial areas of research for the development of an artificial intelligence solution to mitigate the impact of EW on sUAS. Additionally, there are also subsections identifying several training, optimization, and simulation techniques used in this praxis.

Chapter 3 explains the methods used in this praxis. The chapter details the selection and cleaning of three datasets of Russian military vehicles. Additionally, this chapter outlines the methods used to train the artificial intelligence model and highlights the significance of the NVIDIA Jetson Orin Nano as a suitable test platform.

Chapter 4 of this praxis details the experiment results. Additionally, this chapter provides analysis of the research questions and hypotheses. It includes results of several experiments to address the impacts of lighting conditions, power constraints, and memory limitations on the YOLOv8 model's performance on an edge device.

Chapter 5 provides a conclusion of this praxis, a discussion of key insights, and identifies contributions to the academic body of knowledge pertaining to AI/ML. The results of the research and experimentation conducted demonstrates several findings in the application of deep learning for object detection on edge devices. First, the findings of this praxis demonstrate the potential application of CNNs, specifically the YOLOv8m architecture, for deployment on edge devices like the NVIDIA Jetson Orin Nano platform for use in detecting adversary military assets without the need for a human operator to conduct manual visual detection. There are several hypotheses in this praxis tested to determine the potential performance of the YOLOv8m model in visually complex contested military environments. The results of these tests demonstrate only partial success of the model in its ability to generalize with unseen data as well as conduct object detection under simulated conditions with limited lighting and resource limitations.

## 5.3 Contributions to Body of Knowledge

This praxis explores the intersection of several areas of interest within the defense sector including national defense, military strategy, AI research, and edge computing and provides research and insight into key areas pertinent to the further development and application of artificial intelligence. This praxis contributes to the body of knowledge through the evaluation of the deployment of a deep learning AI model on an edge device for the detection of adversary military assets in DDIL environments.

1. The research conducted in this praxis demonstrates the practical implementation of a YOLOv8m model on an NVIDIA Jetson Orin Nano Super Developer Kit to assist in the mitigation of EW effects on sUAS.
2. This praxis provides insight and validation into the use of Unreal Engine environments and objects to simulate real-world environments to test model generalization.
3. The research conducted in this praxis provides insight and validation into hardware limitations and their impact on the YOLOv8m model.

**5.4 Recommendations for Future Research**

The nexus between AI/ML, sUAS, EW, and military applications has created several areas for future research to build upon. One area for future research may be the continued optimization of AI/ML models on edge devices for use on sUAS platforms and the combination with lightweight autonomous navigation systems. Another area for future research may be the integration of additional lightweight models to further augment the detection capabilities of the YOLO architecture to enhance decision making of AI models deployed on edge devices in military applications like ISR or kinetic capabilities. One final area for future research may include further development of image datasets using simulated environments like Unreal Engine. Enhancements to existing datasets or even the creation of new datasets may provide high fidelity imagery in various environmental conditions and viewing angles which may assist in the development of future models.

### 5.5 Ethical Considerations

The implementation of deep learning capabilities on sUAS may enable the potential automated detection of adversary military assets for applications like ISR as well as kinetic strike capabilities when conducting military operations in EW contested environments. Although this technology may present improvements to military capabilities, the deployment of AI for these purposes presents potential ethical concerns. This may especially be the case if these technologies are implemented to surveil or harm indiscriminately. According to International Humanitarian Law as detailed by The International Committee of the Red Cross (2002), military combatants must adhere to several principles to ensure the safety of civilians. These principles include a requirement for the *distinction* between combatants and non-combatants, *proportional* use of force to prevent impact to civilians, *necessity* in the application of force, as well as *limitations* to the use of weapons that cause unnecessary injury or suffering (International Committee of the Red Cross, 2002). In addition to these existing requirements, The International Committee of the Red Cross (2018) also provides insight into the ethics of deploying autonomous weapon systems and enumerates its position in the publication *Ethics and Autonomous Weapon Systems: An Ethical Basis for Human Control*, that control over autonomous weapon systems must be maintained by humans to assist in the prevention of unethical use of these technologies. According to Arkin (2018), there may also be potential safety benefits to the use of lethal autonomous weapon systems (LAWS) over conventional weapons systems as LAWS may have greater degrees of precision and predictability that may potentially limit the impact on non-combatants. The implementation of AI augmented capabilities on sUAS for military applications may

have the potential to provide increased reliability and precision in EW contested environments when compared to non-AI augmented sUAS.

To help prevent unethical use of these technologies, responsible and ethical engineering practices as well as accountability in military decision-making should be ensured. This may include factors like ethical data collection and use, rigorous test and evaluation to validate real-world predictability, human control over use of the autonomous system, and human accountability over the application of these technologies. In addition to these factors, the U.S. Department of Defense enumerated several AI ethical principles guiding the development and use of AI. According to the U.S. Department of Defense (2021), these principles include *responsibility* to ensure adequate judgement is used in the application of AI, *equity* to ensure mitigation of bias in AI systems, *traceability* to ensure auditable methodologies in decision making, *reliability* to ensure safety and security, and *governance* to ensure that AI systems function as intended and can be remotely deactivated. The proliferation of sUAS in military applications along with the potential combination with AI augmentations may likely continue to increase and improve in the future. As sUAS and AI capabilities continue to improve, it is crucial that ethical development and application of these technologies are conducted to ensure that it is done safely, responsibly, and maintains human accountability to safeguard from potential misuse by malign actors.

## References


Abubakar, A., Farah, M., Alsheikh, M., & Saif, A. A. (2023). Optimal Altitude Range for Image Collection Drone from Agricultural Areas with Image Resolution and Power Consumption Constraints. *2022 19th International Multi-Conference on Systems, Signals &Amp; Devices (SSD)*, 689–693. https://doi.org/10.1109/ssd58187.2023.10411193

Alhassan, M. a. M., & Yılmaz, E. (2025). Evaluating YOLOV4 and YOLOV5 for enhanced object detection in UAV-Based surveillance. *Processes*, *13*(1), 254. https://doi.org/10.3390/pr13010254

Ali, M. L., & Zhang, Z. (2024). The YOLO Framework: a comprehensive review of evolution, applications, and benchmarks in object detection. *Computers*, *13*(12), 336. https://doi.org/10.3390/computers13120336

Arezoomandan, S., Klohoker, J., & Han, D. K. (2024). Analyzing the efficacy of synthetic images in unmanned aerial vehicle detection. *2023 IEEE International Conference on Consumer Electronics (ICCE)*, 1–6. https://doi.org/10.1109/icce59016.2024.10444259

Arkin, R. (2018). One nation, under drones: legality, morality, and utility of unmanned combat systems (J. E. Jackson, Ed.). Naval Institute Press.

Arshad, M. A., Khan, S. H., Qamar, S., Khan, M. W., Murtza, I., Gwak, J., & Khan, A. (2022). Drone navigation using region and Edge Exploitation-Based Deep CNN. *IEEE Access*, *10*, 95441–95450. https://doi.org/10.1109/access.2022.3204876

Busch, M. (2018). *Unmanned Aerial Systems Miniaturization: Chances and Risks of an Irreversible Trend* (Journal Edition 25). Joint Air Power Competence Center (JAPCC). https://www.japcc.org/wp-content/uploads/JAPCC_J25_screen.pdf

Çetin, E., Barrado, C., & Pastor, E. (2023). Explainability of Deep Reinforcement Learning Method with Drones. *2013 IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC)*, 1–9. https://doi.org/10.1109/dasc58513.2023.10311156

Chávez, K. (2024, August 15). *SMALL DRONES FOR BIG MILITARIES: THE WAY AHEAD*. War Room - U.S. Army War College. https://warroom.armywarcollege.edu/articles/small-drones/

De Oliveira Barbosa, M., Aguiar, F. P. L., Sousa, S. D. S., Cordeiro, L. D. S., De Alencar Nääs, I., & Okano, M. T. (2025). YOLOv8m for Automated Pepper Variety Identification: Improving Accuracy with Data Augmentation. *Applied Sciences*, *15*(13), 7024. https://doi.org/10.3390/app15137024

Deng, L., Bi, L., Li, H., Chen, H., Duan, X., Lou, H., Zhang, H., Bi, J., & Liu, H. (2023). Lightweight aerial image object detection algorithm based on improved YOLOv5s. *Scientific Reports*, *13*(1). https://doi.org/10.1038/s41598-023-34892-4

Devos, A., Ebeid, E., & Manoonpong, P. (2018). Development of Autonomous Drones for Adaptive Obstacle Avoidance in Real World Environments. *2018 21st Euromicro Conference on Digital System Design (DSD). IEEE*, 707–710. https://doi.org/10.1109/dsd.2018.00009

Ding, B., Zhang, Y., & Ma, S. (2024). A lightweight Real-Time infrared object detection model based on YOLOV8 for unmanned aerial vehicles. *Drones*, *8*(9), 479. https://doi.org/10.3390/drones8090479

Dou, H., Chen, S., Xu, F., Liu, Y., & Zhao, H. (2025). Analysis of vehicle and pedestrian detection effects of improved YOLOv8 model in drone-assisted urban traffic monitoring system. *PLoS ONE*, *20*(3), e0314817. https://doi.org/10.1371/journal.pone.0314817

Farkhodov, K., Lee, S., Platos, J., & Kwon, K. (2023). Deep reinforcement learning TF-Agent-Based object tracking with virtual autonomous drone in a game engine. *IEEE Access*, *11*, 124129–124138. https://doi.org/10.1109/access.2023.3325062

Finer, J., & Shimer, D. (2025, July 15). Ukraine's drone Revolution: And what America should learn from it. *Foreign Affairs*. https://www.foreignaffairs.com/russia/ukraines-drone-revolution

Gerstein, D. M., & Leidy, E. N. (2024, February 15). *Emerging technology and risk analysis: Unmanned aerial systems intelligent swarm technology*. RAND. https://www.rand.org/pubs/research_reports/RRA2380-1.html

Hakani, R., & Rawat, A. (2024). Edge Computing-Driven Real-Time drone detection using YOLOV9 and NVIDIA Jetson Nano. *Drones*, *8*(11), 680. https://doi.org/10.3390/drones8110680

Hambling, D. (2022, April 26). *How heavy are Russian losses, and what does it mean for their offensive?* Forbes. https://www.forbes.com/sites/davidhambling/2022/04/26/how-heavy-are-russian-losses-and-what-does-it-mean-for-their-offensive/

Hammes, T. X. (2021, October 20). *Defence Dominance: Advantage for Small States*. Institute for National Strategic Studies National Defense University.

https://inss.ndu.edu/Media/News/Article/2817074/defence-dominance-advantage-for-small-states/

Han, B., Lee, J., Lim, K., & Choi, D. (2020). Design of a scalable and fast YOLO for Edge-Computing devices. *Sensors*, *20*(23), 6779. https://doi.org/10.3390/s20236779

International Committee of the Red Cross. (2018). Ethics and autonomous weapon systems: An ethical basis for human control. https://www.icrc.org/sites/default/files/document/file_list/icrc_ethics_and_autonomous_weapon_systems_report_3_april_2018.pdf

International Committee of the Red Cross. (2002). The Law of Armed Conflict. https://www.icrc.org/sites/default/files/external/doc/en/assets/files/other/law1_final.pdf

Kallenborn, Z., & Plichta, M. (2024). *Breaking the Shield: Countering Drone Defenses* (Joint Force Quarterly No. 113). National Defense University. https://digitalcommons.ndu.edu/cgi/viewcontent.cgi?article=1004&context=joint-force-quarterly

Krüger, M., Gilbert, D., Kuhlen, T. W., & Gerrits, T. (2024). Game engines for Immersive Visualization: Using Unreal Engine Beyond entertainment. *PRESENCE Virtual and Augmented Reality*, *33*, 31–55. https://doi.org/10.1162/pres_a_00416

Kumar, C., & Jannesari, A. (2023). Optimal Deep Learning model for UAVs: a case study. *2023 26th International Symposium on Wireless Personal Multimedia*

*Communications (WPMC). IEEE*, 51–57. https://doi.org/10.1109/WPMC59531.2023.10338971

Lai, K., Chung, Y., Su, J., Lai, C., & Huang, Y. (2023). AI Wings: an AIoT drone system for commanding ArduPilot UAVs. *IEEE Systems Journal*, *17*(2), 2213–2224. https://doi.org/10.1109/jsyst.2022.3189011

Lee, M., Choi, M., Yang, T., Kim, J., Kim, J., Kwon, O., & Cho, N. (2024). A study on the advancement of intelligent military drones: Focusing on reconnaissance operations. *IEEE Access*, *12*, 55964–55975. https://doi.org/10.1109/access.2024.3390035

Li, H., & Wu, J. (2024). LSOD-YOLOv8s: A Lightweight Small Object Detection Model Based on YOLOv8 for UAV Aerial Images. *Engineering Letters*, *32*(11), 2073–2082.

Ma, M., Shen, S., & Huang, Y. (2023). Enhancing UAV Visual Landing Recognition with YOLO's Object Detection by Onboard Edge Computing. *Sensors*, *23*(21), 8999. https://doi.org/10.3390/s23218999

Manikandan, K., Olayil, R., Govindan, R., & Thangavelu, R. (2024). Artificial Intelligence based Drone Control for Monitoring Military Environment and its Security Applications. *8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2024 - Proceedings. IEEE*, 572–576. https://doi.org/10.1109/i-smac61858.2024.10714824

Micheal, A. A., Vani, K., Sanjeevi, S., & Lin, C. (2020). Object Detection and Tracking with UAV Data Using Deep Learning. *Journal of the Indian Society of Remote Sensing*, *49*(3), 463–469. https://doi.org/10.1007/s12524-020-01229-x

Milonas, D., Knight, T., & Hundman, E. (2024). *Attritable Unmanned Aircraft Systems: Conceptualization and Key Players*. Defense Systems Information Analysis Center (DSIAC). https://dsiac.dtic.mil/wp-content/uploads/2024/11/SOAR_DSIAC_Attritable-Unmanned-Aircraft-Systems-Conceptualization-and-Key-Players_11252024.pdf

Mittal, P. (2024). A comprehensive survey of deep learning-based lightweight object detection models for edge devices. *Artificial Intelligence Review*, *57*(9). https://doi.org/10.1007/s10462-024-10877-1

Molloy, O. (2024). *Drones in Modern Warfare: Lessons Learnt from the War in Ukraine* (Australian Army Occasional Paper No. 29). Australian Army Research Centre. https://doi.org/10.61451/267513

NVIDIA. (2024). *Jetson Orin Nano Datasheet*. https://nvdam.widen.net/s/zkfqjmtds2/jetson-orin-datasheet-nano-developer-kit-3575392-r2

O'Conner, M. (2023, October 19). *Replicator: a bold new path for DOD*. Georgetown: Center for Security and Emerging Technology. https://cset.georgetown.edu/article/replicator-a-bold-new-path-for-dod/

Parmar, A., Gaiiar, R., & Gajjar, N. (2023). Drone based Potholes detection using Machine Learning on various Edge AI devices in Real-Time. *2021 IEEE*

Patterson, D. (2018). *One nation, under drones: legality, morality, and utility of unmanned combat systems* (J. E. Jackson, Ed.). Naval Institute Press.

*International Symposium on Smart Electronic Systems (iSES)*, 22–26. https://doi.org/10.1109/ises58672.2023.00016

Pledger, T. (2021). *The Role of Drones in Future Terrorist Attacks* (Land Warfare Paper 137). The Association of the United States Army (AUSA). https://www.ausa.org/sites/default/files/publications/LWP-137-The-Role-of-Drones-in-Future-Terrorist-Attacks_0.pdf

Qu, J., Li, Q., Pan, J., Sun, M., Lu, X., Zhou, Y., & Zhu, H. (2025). SS-YOLOv8: small-size object detection algorithm based on improved YOLOv8 for UAV imagery. *Multimedia Systems*, *31*(1). https://doi.org/10.1007/s00530-024-01622-3

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. https://doi.org/10.1109/cvpr.2016.91

Rey, L., Bernardos, A. M., Dobrzycki, A. D., Carramiñana, D., Bergesio, L., Besada, J. A., & Casar, J. R. (2025). A performance analysis of you only look once models for deployment on constrained computational edge devices in drone applications. *Electronics*, *14*(3), 638. https://doi.org/10.3390/electronics14030638

Saeed, Z., Yousaf, M. H., Ahmed, R., Velastin, S. A., & Viriri, S. (2023). On-Board Small-Scale object Detection for Unmanned aerial vehicles (UAVs). *Drones*, *7*(5), 310. https://doi.org/10.3390/drones7050310

Samanth, S., K, P., V., & Balachandra, M. (2022). Security in Internet of Drones: A Comprehensive review. *Cogent Engineering*, *9*(1). https://doi.org/10.1080/23311916.2022.2029080

Sanikommu, V., Marripudi, S. P., Yekkanti, H. R., Divi, R., Chandrakanth, R., & Mahindra, P. (2025). Edge computing for detection of ship and ship port from

remote sensing images using YOLO. *Frontiers in Artificial Intelligence*, *8*. https://doi.org/10.3389/frai.2025.1508664

Saydirasulovich, S. N., Mukhiddinov, M., Djuraev, O., Abdusalomov, A., & Cho, Y. (2023). An improved wildfire smoke detection based on YOLOV8 and UAV images. *Sensors*, *23*(20), 8374. https://doi.org/10.3390/s23208374

Sayler, K. (2024). *Emerging Military Technologies: Background and Issues for Congress* (No. R46458). Congressional Research Service. https://www.congress.gov/crs_external_products/R/PDF/R46458/R46458.13.pdf

Shaikh, A. M., & Gajjar, R. (2023). Performance Evaluation of Ambulance Detection and Classification Using Machine Learning Through a Drone on Edge AI Devices. *IEEE Electronic Library (IEL) Conference Proceedings*, 773–778. https://doi.org/10.1109/aece59614.2023.10428611

Shimada, T., Nishikawa, H., Kong, X., & Tomiyama, H. (2022). Pix2Pix-Based Monocular Depth Estimation for Drones with Optical Flow on AirSim. *Sensors*, *22*(6), 2097. https://doi.org/10.3390/s22062097

Terven, J., Córdova-Esparza, D., & Romero-González, J. (2023). A comprehensive review of YOLO architectures in computer vision: from YOLOV1 to YOLOV8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, *5*(4), 1680–1716. https://doi.org/10.3390/make5040083

U.S. Department of Defense. (2023). *Data, Analytics, and Artificial Intelligence Adoption Strategy: Accelerating Decision Advantage*. Department of Defense. https://media.defense.gov/2023/nov/02/2003333300/-1/-1/1/dod_data_analytics_ai_adoption_strategy.pdf

U.S. Department of Defense. (2024). *Deputy Secretary of Defense Kathleen Hicks Announces Additional Replicator All-Domain Attritable Autonomous Capabilities*. https://www.defense.gov/News/Releases/Release/Article/3963289/deputy-secretary-of-defense-kathleen-hicks-announces-additional-replicator-all/

U.S. Department of Defense. (2021). Implementing Responsible Artificial Intelligence in the Department of Defense. https://media.defense.gov/2021/May/27/2002730593/-1/-1/0/IMPLEMENTING-RESPONSIBLE-ARTIFICIAL-INTELLIGENCE-IN-THE-DEPARTMENT-OF-DEFENSE.PDF

U.S. Department of Homeland Security. (2019). *Privacy Impact Assessment for the Small Unmanned Aircraft Systems* (DHS/CISA/PIA-031). https://www.dhs.gov/sites/default/files/publications/privacy-pia-dhscisa031-suas-july2019v2.pdf

Vaddi, S., Kim, D., Kumar, C., Shad, S., & Jannesari, A. (2021). Efficient object detection model for real-time UAV application. *Computer and Information Science*, *14*(1), 45. https://doi.org/10.5539/cis.v14n1p45

Wang, G., Chen, Y., An, P., Hong, H., Hu, J., & Huang, T. (2023). UAV-YOLOV8: A Small-Object-Detection model based on improved YOLOV8 for UAV aerial photography scenarios. *Sensors*, *23*(16), 7190. https://doi.org/10.3390/s23167190

Watling, J., & Reynolds, N. (2023). *Meatgrinder: Russian Tactics in the Second Year of Its Invasion of Ukraine*. Royal United Services Institute for Defence and Security Studies. https://static.rusi.org/403-SR-Russian-Tactics-web-final.pdf

Wu, H., Wei, H., & Wang, H. (2024). Stray Dogs Tracking Based on Drone and AI. *IEEE International Conference on Consumer Electronics-China (Online)*, 205–206. https://doi.org/10.1109/icce-taiwan62264.2024.10674589

Xie, D., Hu, R., Wang, C., Zhu, C., Xu, H., & Li, Q. (2023). A simulation framework of unmanned aerial vehicles route planning design and validation for landslide monitoring. *Remote Sensing*, *15*(24), 5758. https://doi.org/10.3390/rs15245758

Yao, G., Zhu, S., Zhang, L., & Qi, M. (2024). HP-YOLOV8: High-Precision Small Object Detection Algorithm for remote sensing images. *Sensors*, *24*(15), 4858. https://doi.org/10.3390/s24154858

Zhai, X., Huang, Z., Li, T., Liu, H., & Wang, S. (2023). YOLO-Drone: an optimized YOLOV8 network for tiny UAV object detection. *Electronics*, *12*(17), 3664. https://doi.org/10.3390/electronics12173664

Zhan, W., Sun, C., Wang, M., She, J., Zhang, Y., Zhang, Z., & Sun, Y. (2021). An improved Yolov5 real-time detection method for small objects captured by UAV. *Soft Computing*, *26*(1), 361–373. https://doi.org/10.1007/s00500-021-06407-8

Zhang, Y., Niu, J., Huang, Z., Pan, C., Xue, Y., & Tan, F. (2024). High-Precision detection for sandalwood trees via improved YOLOV5s and StyleGAN. *Agriculture*, *14*(3), 452. https://doi.org/10.3390/agriculture14030452

Zhang, Z., Mahmud, M. a. P., & Kouzani, A. Z. (2022). FITNN: a Low-Resource FPGA-Based CNN accelerator for drones. *IEEE Internet of Things Journal*, *9*(21), 21357–21369. https://doi.org/10.1109/jiot.2022.3179016

Zhao, B., Zhou, Y., Song, R., Yu, L., Zhang, X., & Liu, J. (2024). Modular YOLOv8 optimization for real-time UAV maritime rescue object detection. *Scientific Reports*, *14*(1). https://doi.org/10.1038/s41598-024-75807-1

Zhao, X., Zhang, W., Xia, Y., Zhang, H., Zheng, C., Ma, J., & Zhang, Z. (2024). G-YOLO: a lightweight infrared aerial remote sensing target detection model for UAVs based on YOLOV8. *Drones*, *8*(9), 495. https://doi.org/10.3390/drones8090495

# Appendix A

## 1 Oryx 2022 Russia Ukraine War Dataset (Kaggle)

https://www.kaggle.com/datasets/piterfm/2022-ukraine-russia-war-equipment-losses-oryx/data

## 2 Military_Vehicles Dataset (Hugging Face)

https://huggingface.co/datasets/lab-v2/military_vehicles

## 3 War_Tech_2.0 Detection Objects Dataset (Kaggle)

https://www.kaggle.com/datasets/gon213/war-tech-v2-0-by-gontech

# Appendix B

## 1 Link to AI/ML Code Repository (GitHub)

https://github.com/seans-GW/SEAS_2025_Praxis_Appendix_B