**Visually Adaptive Neural Networks for Integrated Synthetic Holographic**

**(VANISH)**

by Gregory P. Lundy

B.S. in Marine Transportation Operations, May 2003, Maine Maritime Academy
M.B.A. in Corporate Finance, May 2013, University of Virginia
M.S. in Systems Engineering, June 2013, The George Washington 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

August 31, 2025

Praxis directed by

Vijay Raghavan
Professorial Lecturer in Engineering and Applied Science

The School of Engineering and Applied Science of The George Washington University certifies that Gregory Patrick Lundy has passed the Final Examination for the degree of Doctor of Engineering as of August 31, 2025. This is the final and approved form of the Praxis.

**Visually Adaptive Neural Networks for Integrated Synthetic Holographic (VANISH)**

Gregory P. Lundy

Praxis Research Committee:

Hamza F. Alsarhan, Professorial Lecturer of Engineering Management and Systems Engineering, Committee Chair

Aaron Focacci, Professorial Lecturer in Engineering and Applied Science, Committee Member

Vijay Raghavan, Professorial Lecturer in Engineering and Applied Science, Committee Advisor

© Copyright 2025 by Gregory P. Lundy
All rights reserved

## Dedication

This Praxis is appreciatively dedicated to my wife, Whitney, my parents, Nicole and James, and my in-laws, Patricia and Michael. Their support, encouragement, and constant love sustained me during this academic endeavor and throughout my life.

## Abstract of Praxis


### Visually Adaptive Neural Networks for Integrated Synthetic Holographic (VANISH)

Technological advancements on the modern battlefield are driving renewed focus for the development and implementation of advanced camouflage techniques to increase the survivability of both manned and unmanned military platforms. The ongoing development of computer vision-aided detection systems and autonomous combatants requires a new approach to camouflage that provides both qualitative and quantitative enhancements beyond traditional static methods (Kim, 2018). This Praxis explores the concept of blending artificial intelligence (AI) and machine learning (ML) algorithms with photonic smart-skin technologies to create a Visually Adaptive Neural-Network for Integrated Synthetic Holographic (VANISH) applications. This research aims to enhance battlefield stealth by leveraging AI/ML techniques to camouflage troops and their associated equipment in real-time. Much as moviegoers witnessed in John McTiernan's 1987 film "Predator", VANISH proposes developing and testing a dynamically adaptive camouflage "smart skin" that responds to visual changes in the environment (Wikipedia, 2024).

If operationalized, VANISH technology may address the limitations of static camouflage by incorporating AI/ML adaptability. Traditional camouflaging techniques rely on pre-designed patterns that may not be effective in varying environments, as may be found in urban-based conflicts (Brimley, 2014). This Praxis proposes using deep learning algorithms to continuously analyze the surrounding environments and generate synthetic patterns that seamlessly blend with the

background. This real-time adaptability is crucial in modern all-domain combat scenarios where the environments and fighting conditions can shift rapidly.

The deliberate use of a wide range of AI/ML libraries and techniques was researched and chosen to allow VANISH to scale across different levels of complexity, from simple color matching applications to sophisticated projections. This scalability would ensure the system can be deployed in various operational contexts, from reconnaissance missions to full-scale combat scenarios. As military operations become increasingly more complex, the ability to integrate and operationalize sophisticated AI/ML applications will undoubtedly play a pivotal role in maintaining strategic advantages over sophisticated adversaries (Scharre, 2018).

In conjunction with AI/ML applications, the VANISH Praxis also explores the emerging world of flexible smart skin materials and their potential role in delivering the rendered camouflage outputs from the VANISH baseline code onto a military platform. These smart skins serve as the optical interface between the end user and the VANISH codebase and play a pivotal role in demonstrating the tactical intersection between advanced AI/ML technologies and the all-domain war fighter.

As the scrutiny of AI/ML applications grows in the defense sector, this Praxis also explores the critical ethical and legal considerations around this capability. As adaptive camouflage technologies like VANISH advance, concerns around their use in deception, the escalation of conflicts, and the potential for misuse in non-military contexts are sure to arise (Lin, 2019).

This Praxis concludes by making recommendations for future areas of research that bridge the gap between electro-optic (EO), infrared (IR), and radio frequency (RF)

methods of camouflage to create a single unified solution based on the VANISH baseline.

## Table of Contents

Cover Page ..... i

Dedication ..... iv

Abstract of Praxis ..... v

List of Figures ..... xii

List of Tables ..... xv

List of Acronyms ..... xvi

1.1 Background ..... 1

1.2 Research Motivation ..... 2

1.3 Problem Statement ..... 3

1.4 Thesis Statement ..... 4

1.5 Research Objectives ..... 4

1.6 Research Questions and Hypotheses ..... 4

1.7 Scope of Research ..... 5

1.8 Research Limitations ..... 7

1.9 Organization of Praxis ..... 7

Chapter 2—Literature Review ..... 9

2.1 Introduction ..... 9

2.2 The Adobe Primrose Project ..... 10

2.3 Adaptive Optoelectronic Camouflage Systems with Designs Inspired by Cephalopod Skins ..... 12

2.4 Adaptive Camouflage in Defense: Science Behind Real-Time Color-Changing Tech ..... 16

2.5 Summary and Conclusion ..... 17

Chapter 3—Methodology ..... 19
3.1 Introduction ..... 19
3.2 AI/ML Algorithm and Baseline Code Development Stage ..... 19
3.2.1 Applied AI/ML Algorithms for Achieving Adaptive Camouflage ..... 20
3.2.2 Applied AI/ML Algorithms for Camouflage Detection ..... 29
3.3 Materials Science Evaluation Stage ..... 36
3.3.1 Photochromic Materials – Spiropyran ..... 37
3.3.2 Liquid Crystal-Based Smart Skins (LC) ..... 40
3.3.3 Quantum Dot Display (QDD) Smart Skins ..... 43
3.4 Lab and Field Testing Stage ..... 47
3.4.1 Experimentation Design ..... 47
3.4.2 Simulation Environment ..... 48
3.4.3 Flexible LED Smart Skin ..... 50
3.4.5 Integrated VANISH Design Framework for Simulation, Lab, and Field Testing ..... 51
Chapter 4 – Results ..... 54
4.1 Introduction ..... 54
4.2 VANISH AI/ML Code Development ..... 55
4.2.1 VANISH AI/ML Code Development Timeline ..... 56
4.2.2 VANISH CNN Model Training Attempt ..... 57
4.2.3 VANISH Scikit-Learn + SciPy Model for Adaptive Camouflage ..... 59
4.2.4 VANISH OpenCV + Scikit-Image Model for Adaptive Camouflage ..... 60
4.2.5 VANISH OpenCV Model for Adaptive Camouflage ..... 62
4.2.6 VANISH OpenCV Model Initial Graphical User Interface Design ..... 64
4.2.7 VANISH AI/ML Model Final ..... 65

4.3 Lab Test of VANISH Code ..... 68
4.3.1 Testing Framework and Evaluation ..... 69
4.3.2 Lab Test Results ..... 69
4.3.3 Limitations and Challenges Identified ..... 71
4.3.4 Conclusion and Next Steps ..... 72
4.4 VANISH AI/ML Code Field Tests ..... 73
4.4.1 Field Test Employment ..... 74
4.4.1.1 Beer Can Island (Maritime & Desert) Field Test Results ..... 75
4.4.1.2 Beer Can Island Imagery Analysis ..... 76
4.4.1.3 Beer Can Island Test Summary ..... 78
4.4.2.1 Publix Supermarket (Urban) Field Test Results ..... 78
4.4.2.2 Publix Supermarket Imagery Analysis ..... 80
4.4.2.3 Publix Supermarket Test Summary ..... 81
4.4.3.1 Bull Frog Creek Park (Fields & Forests) Field Test Results ..... 81
4.4.3.2 Bull Frog Creek Park Imagery Analysis ..... 83
4.4.3.3 Bull Frog Creek Park Test Summary ..... 84
4.4.4 Field Test Limitations and Challenges ..... 84
4.4.5 Conclusions and Next Steps ..... 85
Chapter 5—Discussion and Conclusions ..... 86
5.1 Discussion ..... 86
5.1.1 Dual-use Concerns and Potential Misuse ..... 87
5.1.2 Privacy and Surveillance Implications ..... 88
5.1.3 Legal and Regulatory Considerations ..... 89
5.2 Conclusions ..... 90
5.3 Contributions to Body of Knowledge ..... 91

5.4 Recommendations for Future Research ..... 93

References ..... 95

Appendix A ..... 106

A.1 Introduction ..... 106

A.2 VANISH Code Implementation Workflow ..... 119

## List of Figures

Figure 1: VANISH Praxis Research Scope Objectives ..... 6

Figure 2: Project Primrose Live Demo (Adobe Project Primrose, 2024) ..... 10

Figure 3: Project Primrose Smart Fabrics (Adobe Project Primrose, 2024) ..... 11

Figure 4: Light-Programmable Meta Devices (Yu C. , 2014) ..... 13

Figure 5: Cephalopod Skin Inspired Adaptive Camouflage (Yu C. , 2014) ..... 14

Figure 6: Visual and Thermal Responsive Smart Skins (Yu C. , 2014) ..... 15

Figure 7: Adaptive Camouflage Textiles for NextGen Tanks (Businessner, 2023) ..... 16

Figure 8: GANs for SAR to EO Translation SN6-SAROPT Dataset (Qing, 23) ..... 23

Figure 9: CNN Applications in Advanced Military Camouflage (Xiao, 2020) ..... 25

Figure 10: K-Means for Camouflage Design (Minhao Wan, 2022) ..... 26

Figure 11: Gaussian Blur Effects on Original Input Stream (Computer Hope, 2024) . 27

Figure 13: VANISH Metrics and Applied Entropy to Measure Pattern Complexity ... 28

Figure 14: GLCM for Texture Classification (Scikit Image, 2024) ..... 29

Figure 15: Fourier Transform in Adaptive Camouflage CamoFourier (Minh, 2023) .. 31

Figure 16: Results of Object Detection and Optical Flow Using Kalman Filters ..... 33

Figure 18: Principal Component Analysis for Feature Classification (Semantic) ..... 34

Figure 19: YOLO Bounding Boxes and Confidence Class Probability (Joseph, 25) .. 35

Figure 20: MOG2 For Background Detection ..... 36

Figure 23: Spiropyran Molecular Makeup (Rad, 2022) ..... 37

Figure 24: Spiropyran Color Changing Abilities (Mandal, 2022) ..... 39

Figure 25: Real-world Applications of LCs (Bisoyi H. K., 2021) ..... 41

Figure 27: LCs in Applied Window Transitions (Bisoyi H. K., 2021) ..... 42

Figure 28: Synthesis and Assembly of QDDs (Agarwal, 2023) ..... 45

Figure 29: Quantum Dot Display (Jang, 2010) ..... 45
Figure 30: VANISH Testing Process ..... 48
Figure 31: HoloscapeLED M2 Surface Mounted Series (M2 Panel, 2024) ..... 50
Figure 32: VANISH System Level Framework ..... 53
Figure 33: VANISH Code Development Timeline ..... 56
Figure 34: VANISH Code Attempting to Train a CNN ..... 58
Figure 35: Generated Camouflage Output ..... 60
Figure 36: VANISH OpenCV +Scikit-Image Model First K-Means MultiCam ..... 62
Figure 37: VANISH OpenCV Model First PredCam Output ..... 63
Figure 38: Initial VANISH GUI Output ..... 65
Figure 39: Final VANISH GUI and Main Camouflage Renderings ..... 67
Figure 40: VANISH Metrics Dashboard Output ..... 68
Figure 41: Lab Test of the VANISH Framework ..... 69
Figure 42: VANISH Lab Test Video Output from VANISH ..... 72
Figure 43: HoloscapeLED Engineer (M2 Panel, 2024) ..... 72
Figure 44: VANISH Field Testing Locations (Microsoft, 2025) ..... 74
Figure 45: DJI Mini 3 for VANISH Video Input (DJI Corporation, 2025) ..... 74
Figure 46: DJI Mini Video Overwatch Architecture with MQ-58 Valkyrie Model ..... 75
Figure 47: Beer Can Island MultiCam (Desert) Code Processing Metrics ..... 77
Figure 48: Beer Can Island MultiCam (Maritime) Code Processing Metrics ..... 77
Figure 49: Beer Can Island (Maritime) VANISH MultiCam Output ..... 78
Figure 50: Beer Can Island (Desert) VANISH MultiCam Output ..... 78
Figure 51: Publix Supermarket MultiCam (Urban) Code Processing Metrics ..... 80
Figure 52: Publix Super Market (Urban) VANISH MultiCam Output ..... 81
Figure 53 Publix Bull Frog Creek Park MultiCam (Field) Code Processing Metrics 83

Figure 54: Bull Frog Creek VANISH MultiCam Output ..... 84

Figure 55: AI Dual-Use Concerns Near Peer (ASPI Initiative, 2020) ..... 87

Figure 57: VANISH Unified Camouflage System Code Framework ..... 119

## List of Tables

Table 1: Department of Defense Priorities for Camouflage/Concealment/Decoy ..... 2

Table 2: Summary of the AI/ML Libraries to Research for VANISH Codebase ..... 6

Table 3: Summary of the Smart Skin Applications Researched for VANISH ..... 6

Table 4: Summary of the Algorithms Explored in VANISH ..... 20

Table 5: VANISH Software Development Tools and Platforms ..... 49

Table 6: HoloscapeLED M2 Specifications (M2 Panel, 2024) ..... 51

Table 7: VANISH CNN Model Overview ..... 57

Table 8: VANISH Scikit-Learn + SciPy Model Overview ..... 59

Table 9: VANISH OpenCV +Scikit-Image Model Overview ..... 61

Table 10: VANISH OpenCV Model Overview ..... 62

Table 11: VANISH Graphical User Interface (GUI) Model Overview ..... 64

Table 12: VANISH AI/ML Model Final Overview ..... 66

Table 13: VANISH Lab Test Results for Code ..... 70

Table 14: Field Test Site Locations ..... 73

Table 15: VANISH Lab Test Results for Beer Can Island ..... 75

Table 16: VANISH Lab Test Results for Publix Super Market ..... 79

Table 17: VANISH Lab Test Results for Bull Frog Creek Park ..... 81

Table 18: Summary of VANISH Impacts on Military Operations ..... 92

## List of Acronyms

| | |
|---|---|
| 2D | Two-Dimensional |
| 3D | Three-Dimensional |
| AEC | Atomic Energy Commission |
| AI/ML | Artificial Intelligence/Machine Learning |
| AMCU | Australian MultiCam Camouflage Uniform |
| API | Application Programming Interface |
| ASPI | Australian Strategic Policy Institute |
| BGR | Blue, Green, Red |
| BMP | Bitmap |
| CAD | Computer Aided Design |
| CIS | Camouflaged Instance Segmentation |
| CLAHE | Contrast Limited Adaptive Histogram Equalization |
| CMOS | Complementary Metal–Oxide–Semiconductor |
| CNAS | Center for a New American Security |
| CNN | Convolutional Neural Networks |
| COD | Camouflaged Object Detection |
| CPU | Central Processing Unit |
| CSS | Cascading Style Sheets |
| CSV | Comma-Separated Values |
| CVD | Chemical Vapor Deposition |
| DOD | Department of Defense |
| DOI | Digital Object Identifier |

| | |
|---|---|
| DPM | Disruptive Pattern Material |
| DQN | Deep Q-Networks |
| EO | Electro-Optical |
| EU | European Union |
| FBI | Federal Bureau of Investigation |
| FMV | Full Motion Video |
| FOS | Fiber Optic Sensor |
| FOV | Field of View |
| FPGA | Field-Programmable Gate Array |
| FPS | Frames Per Second |
| FT | Fourier Transform |
| GAN | Generative Adversarial Networks |
| GDPR | General Data Protection Regulation |
| GPU | Graphics Processing Unit |
| GUI | Graphical User Interface |
| HDMI | High-Definition Multimedia Interface |
| HM | Hydrothermal Synthesis Method |
| HTML | HyperText Markup Language |
| HTTP | HyperText Transfer Protocol |
| HTTPS | HyperText Transfer Protocol Secure |
| IDE | Integrated Development Environment |
| IEEE | Institute of Electrical and Electronics Engineers |
| IoT | Internet of Things |
| IOU | Intersection Over Union |

| | |
|---|---|
| JAG | Judge Advocate General |
| JPEG | Joint Photographic Experts Group |
| JSON | JavaScript Object Notation |
| KF | Kalman Filtering |
| LAB | Lightness, A, B (color space) |
| LC | Liquid Crystal |
| LCSS | Lightweight Camouflage Screening System |
| LED | Light-Emitting Diode |
| MARPAT | Marine Pattern |
| MDP | Markov Decision Process |
| MIT | Massachusetts Institute of Technology |
| MP4 | MPEG-4 Part 14 |
| MPEG | Moving Picture Experts Group |
| MQ-58 | Military Designation for Valkyrie Drone |
| MQ-9 | Military Designation for Reaper Drone |
| NATO | North Atlantic Treaty Organization |
| OCP | Operational Camouflage Pattern |
| OS | Operating System |
| PCA | Principal Component Analysis |
| PDF | Portable Document Format |
| PDLC | Polymer Dispersed Liquid Crystal |
| PEDOT | Poly(3,4-ethylenedioxythiophene) |
| PNG | Portable Network Graphics |
| PPO | Proximal Policy Optimization |

| | |
|---|---|
| QDD | Quantum Dot Display |
| RAM | Random Access Memory |
| R-CNN | Regional Convolutional Neural Network |
| RGB | Red, Green, Blue |
| RH | Relative Humidity |
| ROE | Rules of Engagement |
| ROEE | Rules of Employment |
| RoI | Region of Interest |
| RPN | Region Proposal Network |
| SAR | Synthetic Aperture Radar |
| SDK | Software Development Kit |
| SGM | Sol-Gel Processing Method |
| SIPRI | Stockholm International Peace Research Institute |
| TTP | Tactics, Techniques and Procedures |
| UAV | Unmanned Aerial Vehicle |
| UCP | Universal Camouflage Pattern |
| UCS | Unified Camouflage System |
| UK | United Kingdom |
| URL | Uniform Resource Locator |
| US | United States |
| USB | Universal Serial Bus |
| USENIX | Advanced Computing Systems Association |
| UV | Ultraviolet |
| VANISH | Visually Adaptive Neural-Networks for Integrated Synthetic Holographic |

| | |
|---|---|
| Wi-Fi | Wireless Fidelity |
| WO3 | Tungsten Trioxide |
| WT | Wavelet Transform |
| XLS | Excel Legacy Format |
| XLSX | Excel Spreadsheet Format |
| YOLO | You Only Look Once |

# Chapter 1—Introduction

## 1.1 Background

Concealment from enemy forces has been a fundamental military objective throughout history. Camouflage techniques have served as part of the war-fighting toolkit for centuries, with origins tracing back to ancient civilizations. Early military leaders recognized that blending with the environment provided significant tactical advantages that allowed forces to surprise opponents and in doing so gain a strategic positioning on the battlefield. Historians have traced camouflage back to ancient Greek and Roman times, when soldiers used mud and foliage to conceal themselves. In World War I, both the Axis and Allied forces introduced multi-colored uniforms and netting in the hopes of further concealing their forces. World War II cemented the concept of camouflage, both on and off the front lines, as the development of radar systems and overhead imagery began to significantly extend the range of military operations (Bar-Shalom, 2011).

Modern military camouflage continues to prove itself a crucial aspect of warfighting strategy. New techniques involve using advanced patterns, colors, and materials that simultaneously blend with multiple domains. Advanced camouflage now includes multispectral designs that can deceive the human eye and confuse infrared, thermal, and radar detection systems. What was once thought to be only science fiction is now becoming a reality. Adaptive camouflage technologies, such as smart fabrics and meta-surface applications, are being developed to dynamically alter appearance in real-time, further enhancing stealth capabilities on the battlefield (Businessner, 2023).

### 1.2 Research Motivation

As discussed previously, traditional camouflage methods are becoming less effective against modern surveillance systems. With AI/ML techniques being used to increase the lethality of modern weapon systems, so too might AI/ML be used to help conceal forces from enemy sensors and troops. The US Department of Defense is actively exploring technologies and prototypes in this venue that can blend AI/ML with new materials to achieve these desired outcomes.

These techniques will hopefully drive tectonic changes in military doctrine that:

1. Enhance operational effectiveness
2. Protect personnel and assets
3. Provide a tactical advantage in modern combat environments

Table 1.0 categorizes the US Department of Defense's priorities and techniques regarding advanced camouflage technologies and their importance on the battlefield.

*Table 1: Department of Defense Priorities for Camouflage/Concealment/Decoy (Army, 1999)*

| Priorities | Optical Techniques | Thermal Techniques | Radar Techniques |
|---|---|---|---|
| **Hiding** | Earth cover, Earth embankments, Vegetation, LCSS (Lightweight Camouflage Screening System), Nets, Screens, RAM (Radar Absorbing Material), Smoke | Earth cover, Earth embankments, Vegetation, LCSS, Thermal paint, Foam, Air conditioning/heating | Earth cover, Earth embankments, Vegetation, LCSS, RAM, Smoke |
| **Blending** | Paint, Vegetation, LCSS, Lights | Thermal paint, Foam, Vegetation, LCSS, Air conditioning/heating | Vegetation, RAM, LCSS, Chaff |
| **Disguising** | Reshaping, Vegetation, Paint, LCSS | Reshaping, Textured mats, LCSS, Insulation | Paint, LCSS, RAM, Corner reflectors, Textured mats, Water |

| | | | |
|---|---|---|---|
| **Disrupting** | Camouflage sails, Smoke, FOS (Fiber Optic Sensor), Flares, Chaff, Pyrotechnics, Corner reflectors, Smudge pots, Balloons, Strobe lights | Smoke, Flares, Tracer simulators | Smoke, Flares |
| **Decoying** | Decoy target (pneumatic or rigid structures), Lights | Decoy target, Air conditioning/heating | Decoy target, Corner reflectors, Smoke, Signal generators |

*Note 1: The priorities listed above are subjective and based on general assumptions, as the actual importance may vary depending on specific military contexts and operations.*

### 1.3 Problem Statement

*The Second Nagorno-Karabakh war and the Ukraine-Russian conflicts have shown that advanced surveillance technologies through drones, satellites, and computer-aided sensors are making traditional military camouflage ineffective, creating an urgent need for adaptive concealment systems that can counter modern detection methods and increase platform survivability.*

The Ukraine-Russia conflict has increased the demand for enhanced military camouflage due to the sophistication of surveillance and reconnaissance technologies. Both sides have employed drones (commercial and military grade), satellite imagery, and electronic warfare systems in their quest for military superiority. Through the correct application of these technologies, both adversaries have significantly diminished the effectiveness of traditional camouflage techniques. As battlefields expand further into multi-domain operations (air, land, sea, and space), and combatants expand their use of connected networks and sensors, traditional camouflage techniques are becoming increasingly irrelevant (Kim, 2018). Therefore, the need for advanced camouflage that can adapt to various environments and counter

emerging threats/sensors beyond traditional camouflage means has become critical for all modern militaries.

### 1.4 Thesis Statement

*AI/ML models that enable real-time visual reflectivity of their environments and deliver synthetic military camouflage are critical to enhancing the security of wartime combatants on the modern battlefield.*

The VANISH Praxis is being developed to study the intersections between AI/ML data fusion and advanced material science. In this Praxis, the use of deep learning methods as well as synthetic data generation techniques will be explored in the hopes of creating an adaptive camouflage system that can readily defeat human and computer vision detection techniques. VANISH looks to merge AI algorithms with photonic smart-skin materials to achieve real-time environmental adaptive camouflage.

### 1.5 Research Objectives

1. Explore the intersections between AI/ML applications and flexible smart skin technologies for use as adaptive camouflage on military platforms.
2. Develop a lightweight AI/ML model that can be used to conduct a series of experiments to determine if the intersection of these two technologies might be viable for employment on the modern battlefield.

### 1.6 Research Questions and Hypotheses

The following research questions and their associated hypotheses were crafted to guide the VANISH research throughout the Praxis:

**RQ1:** Can AI/ML algorithms be used to develop adaptive camouflage models that can counter both human and computer vision detection?

**RQ2:** What types of photonic smart-skin materials can best be paired with AI/ML models to create advanced camouflage for military applications?

**RQ3*:*** What are potential adaptive camouflage technology applications and their associated ethical impacts across modern combat environments?

**H1:** AI/ML models and their associated algorithms can create adaptive camouflage patterns that are more effective and convincing than traditional camouflage patterns in evading human observers and computer vision systems across controlled experimental conditions.

**H2:** Photonic smart-skin materials with high flexibility, rapid response times, and tuneable optical properties (such as dynamic color changes or light-scattering effects) may demonstrate superior compatibility with AI/ML algorithms in creating adaptive camouflage capable of real-time environmental blending in military settings.

**H3:** Ethical challenges may emerge as modern militaries integrate adaptive camouflage technologies into military operations, leading to privacy concerns, issues in data collection, and the potential for dual-use applications.

### 1.7 Scope of Research

The scope of the research contained within the VANISH Praxis is designed to ensure that a multitude of AI/ML techniques and smart skin applications are considered as part of the VANISH Praxis in order to deliver the most realistic, integrated, and resilient adaptive camouflage effects.

The research will support four primary objectives (see Figure 1):

1. VANISH AI/ML Model development (see Table 2)
2. EO-Sensor Integration for Live-Full Motion Video (FMV)
3. Flexible Smart Skin Applications (see Table 3)
4. Demonstrated EO Sensor Outputs

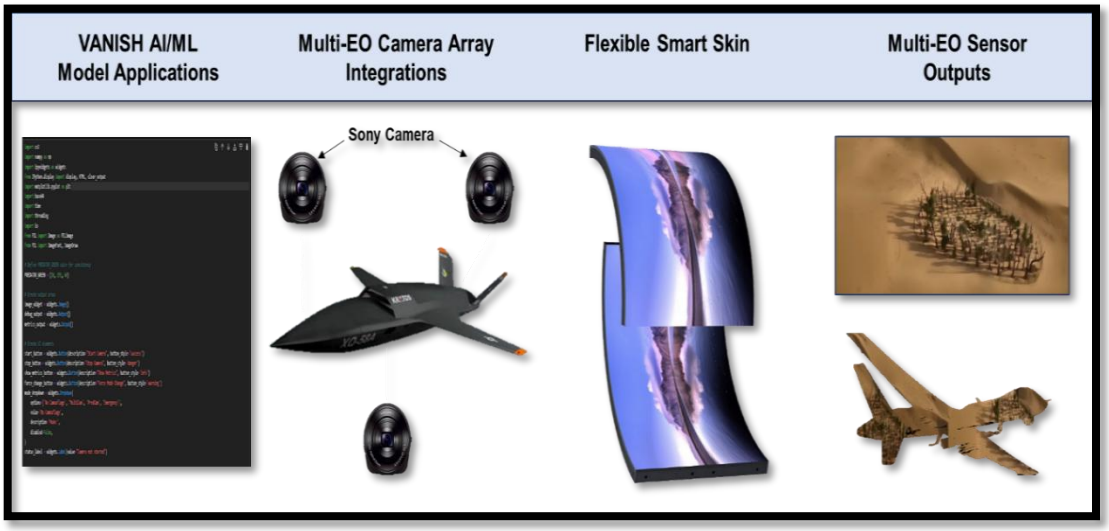



*Figure 1: VANISH Praxis Research Scope Objectives*

*Table 2: Summary of the AI/ML Libraries to Research for VANISH Codebase*

| AI/ML Library | AI/ML Function Name |
|---|---|
| **TensorFlow/ Keras** | Generative Adversarial Networks (GAN) |
| | Convolutional Neural Networks (CNN) |
| | Regional Convolutional Neural Networks (R-CNN) |
| **OpenCV** | K-Means Clustering |
| | Gaussian Mixture Model (MOG2) |
| | Gaussian Blur |
| | Entropy |
| **Scikit-Image** | Wavelet Transform |
| | K-Means Clustering |
| | Gray-Level Co-occurrence Matrix |

*Table 3: Summary of the Smart Skin Applications Researched for VANISH*

| Smart Skin Application | Purpose |
|---|---|
| **Tungsten Trioxide (WO3)** | Blue or grey flexible smart skin |
| **Poly(3,4-ethylenedioxythiophene) or PEDOT** | Light and dark blue tones prone to degradation smart skin |

| | |
|---|---|
| **Spiropyran** | Multiple stimuli smart skin light, temperature, and pressure |
| **Liquid Crystal Base Smart Skins (LC)** | On/Off Smart Skin prone to temperature |
| **Quantum Dot Display (QDD)** | Costly, wide-ranging color palette smart skin |
| **Flexible Light Emitting Diode (LED)*** | Flexible smart skin that serves as the basis for VANISH |

**Note: Lab and field testing were limited to a single flexible LED smart skin configuration produced by HoloscapeLED.*

### 1.8 Research Limitations

Only static based ground testing was conducted as part of this Praxis. Flexible LED smart skins were procured and evaluated for their color palette matching to the baseline VANISH code, but they were never affixed to a military platform. A model MQ-58 Valkyrie drone was CAD rendered, and 3D printed as part of the Praxis design phase, with plans to mount the smart skins to it, but due to limitations in fuselage integrity, the necessary computing hardware components (e.g. Raspberry Pi) could not be integrated onto the platform to support the LED panels.

### 1.9 Organization of Praxis

The VANISH Praxis adheres to the traditional academic thesis structure, consisting of an introduction and justification for research, followed by substantive chapters, and a technical appendix. This document aims to deliver a logical and comprehensive progression for future researchers to follow and replicate in their own testing environments.

The Praxis chapters are structured as follows:

1. **Chapter 1 - Background**

Provides historical context on military camouflage, identifies the core research problem and questions, explains the motivation and significance of the study, and introduces the VANISH approach.

2. **Chapter 2 – Literature Review**

   This chapter analyzes key related projects, including Adobe's Project Primrose, bio-inspired adaptive camouflage, and recent defense industry advancements.

3. **Chapter 3 - Methodology**

   Outlines the research framework for system development, and the methodological approach used to design and test the VANISH system.

4. **Chapter 4 – Results**

   Explores the results from iterative designs of the VANISH code base as well as laboratory and field test results.

5. **Chapter 5 – Discussion and Conclusion**

   Examines the ethical implications and potential concerns associated with advanced military camouflage technologies. Summarizes the VANISH Praxis, outlining its significance, potential impacts, and directions for future research.

**Appendix A: VANISH Code Base for Future Research Reference**

Delivers the applied code and implementation details for the VANISH simulation.

# Chapter 2—Literature Review

## 2.1 Introduction

Adaptive camouflage technologies have made significant progress recently, thanks to advances in computing power and materials science working together. Both private industry and universities have made important breakthroughs in developing materials that can change and adapt in near real time, taking cues from creatures such as the cuttlefish. Three notable projects explored in the pursuit of VANISH applications include:

1. *Adobe's Project Primrose:* This 2024 project showcased for the world the feasibility of integrating flexible electrophoretic displays into wearable fashion (Adobe Project Primrose, 2024). This project, while primarily focused on consumer applications, highlights the potential for adaptive display technologies in various fields, including military camouflage.
2. *Adaptive Optoelectronic Camouflage Systems with Designs Inspired by Cephalopod Skins:* This 2014 research project demonstrated a bio-inspired approach to adaptive camouflage, mimicking the color-changing abilities of cephalopods (e.g. cuttlefish). Their approach utilized ultrathin, flexible arrays of LEDs and light sensors to detect and replicate background patterns rapidly, offering potential applications across both visible and thermal camouflage (Yu C. , 2014).
3. *Adaptive Camouflage in Defense:* The Businessner Editorial Team covers recent project advancements in adaptive camouflage technology across the defense sectors. The team explores various sensors, algorithms, and advanced

materials that industry is experimenting with to achieve real-time environmental adaptation (Businessner, 2023).

These developments across both commercial and defense sectors, represent a significant leap forward in establishing what the future could be for adaptive camouflage technology. The intersection between consumer-oriented and defense-focused technologies has long been an area of focus and may accelerate advancements in both fields leading to technological breakthroughs for both the public and private sectors.

### 2.2 The Adobe Primrose Project

Project Primrose, which was developed by Adobe Research, was designed to explore and demonstrate the fusion of wearable fashion and smart skin technologies. The project, at its core, demonstrated how a dress embedded with flexible electrophoretic displays was able to change patterns and colors in real-time (see Figure 2) (Adobe Project Primrose, 2024). The project showcases the potential for wearable technology to revolutionize personal expression and potentially lead to adaptive wearable camouflage. At the heart of Project Primrose is the use of E Ink technology which is commonly found in e-readers (Albright, 2023). By incorporating E Ink tech into the garment as “textile scales” it allowed the wearer a means of altering the dress's

*Figure 2: Project Primrose Live Demo (Adobe Project Primrose, 2024)*

appearance to meet the personal style of the consumer. Demos of the dress showed that it could display various patterns, designs, and even animations through a smartphone app (Adobe Project Primrose, 2024).

*Figure 3: Project Primrose Smart Fabrics (Adobe Project Primrose, 2024)*

As noted by Christine Dierk, a research scientist at Adobe, "*Project Primrose explores the intersection of technology and style, opening up a whole new area of research and innovation.*" (Albright, 2023). This intersection could lead to new applications such as advanced camouflage, adaptive signage (which is already mainstream), and interactive art installations.

While Project Primrose is primarily a proof of concept, it does highlight Adobe's commitment to expanding beyond its traditional software focus, which many large tech companies are focused on achieving. Broader industry trends towards the Internet of Things (IoT) and smart fabrics are critical in advancing technologies like VANISH, which depends significantly on this advancement. As wearable displays become more prevalent and adopted into military applications, policymakers must address issues such as how the data is collected and the potential for digital divides in fashion and applicability in military applications.

Project Primrose is an excellent example of the intersection between technology and fashion. As noted by Paul Asente an Adobe researcher involved in the

project, *"we're just scratching the surface of what's possible with this technology"* and future researchers should capitalize on this success to drive further alignment between military camouflage technologies and consumer-oriented adaptive fashion (Albright, 2023).

If applied to VANISH, the durability and adaptability that would need to be expounded upon for military applications could also benefit the robustness of Project Primrose's wearable displays. This intersection highlights the potential for cross-pollination between defense-oriented and consumer-focused technologies, where both fields could benefit from coordinated “learning and information sharing,” opening up new avenues for innovation in both fields.

### 2.3 Adaptive Optoelectronic Camouflage Systems with Designs Inspired by Cephalopod Skins

In 2014, researchers from Harvard University conducted a study that took a detailed look into the field of adaptive camouflage technology, drawing inspiration from cephalopods (Yu C. , 2014). Throughout the course of the study, researchers developed an innovative ‘optoelectronic system’ that mimics the adaptive camouflage mechanisms found in cephalopod skins, particularly those of octopuses and cuttlefish. This bio-inspired approach is theorized that it may one day create artificial ecosystems capable of rapid color changes to blend with their surroundings. In the course of the study, the researchers developed a network of flexible arrays using light-emitting diodes (LEDs) and light sensors called Light-Programmable Meta Devices (see Figures 4 and 5) as the artificial cuttlefish skin (Yu C. , 2014). These arrays were integrated in a multi-layer structure that included a silver layer serving as a reflective base, similar to the light-reflecting cells found in squid and octopus skin. The

experiment showed that the system could detect and replicate background patterns with accuracy (Yu C. , 2014).

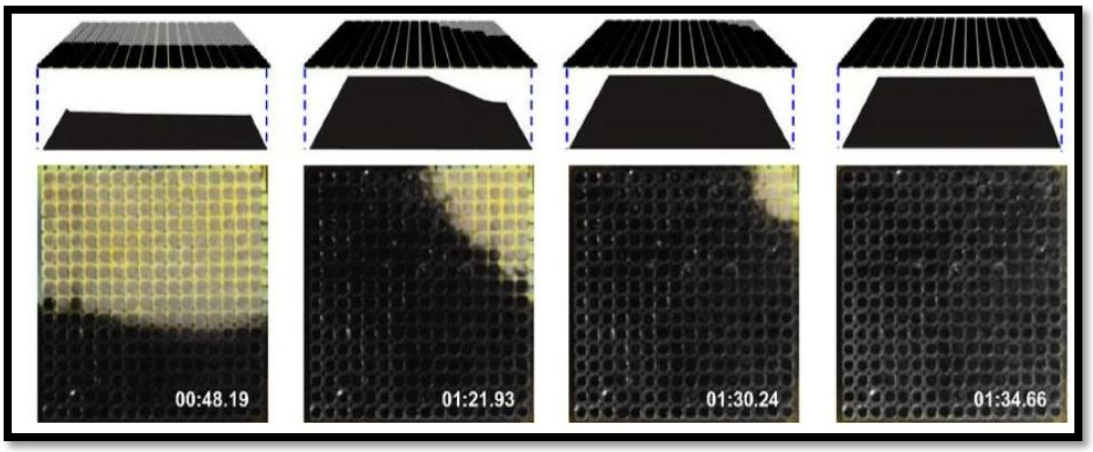



*Figure 4:* Light-Programmable Meta Devices (Yu C. , 2014)

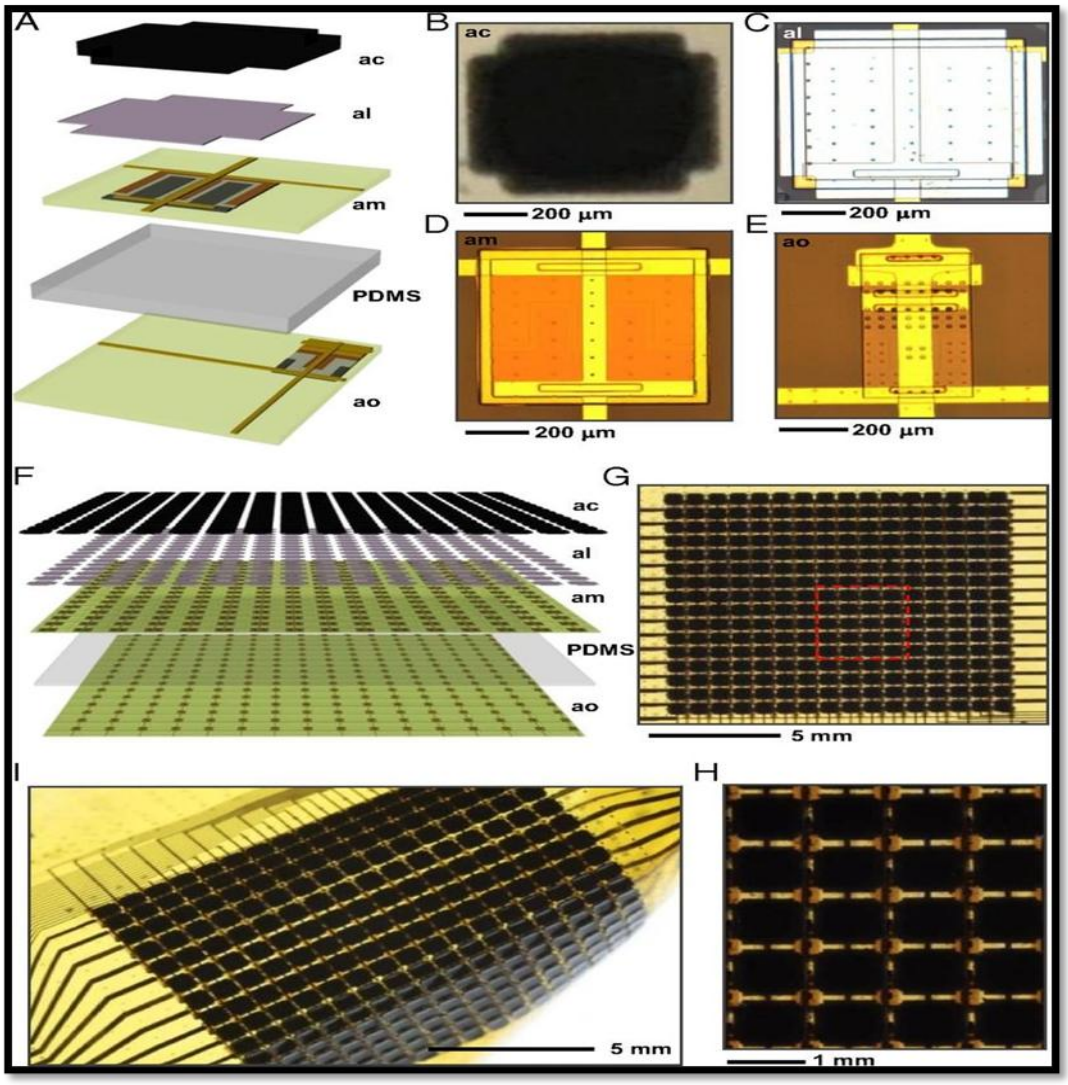



*Figure 5: Cephalopod Skin Inspired Adaptive Camouflage* (Yu C. , 2014)

While the study primarily focuses on visual concealment, the team also explored the potential for thermal camouflage. By incorporating a layer capable of producing heat, the researchers demonstrated the possibility of creating a system that could adapt both visually and thermometrically (see Figure 6) (Yu C. , 2014). This multi-modal approach to camouflage represents a significant step towards creating

truly comprehensive adaptive camouflage ecosystems.

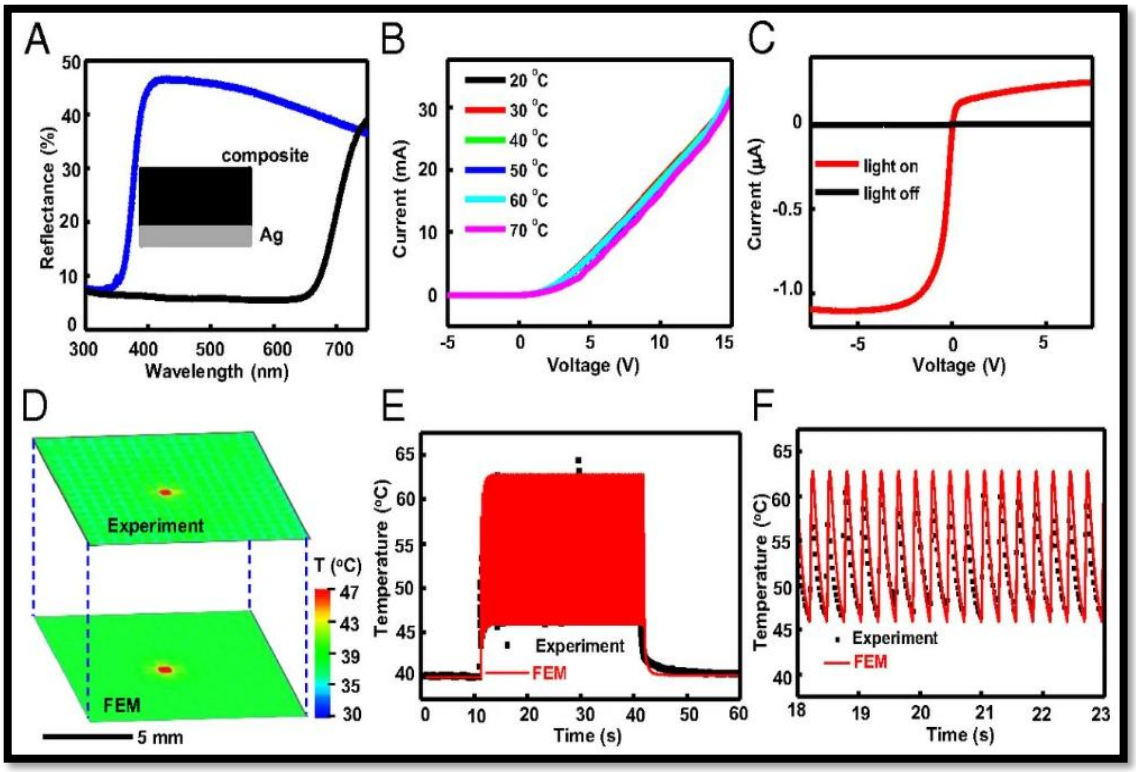



*Figure 6: Visual and Thermal Responsive Smart Skins* (Yu C. , 2014)

The implications of this research extend far beyond military applications. The team suggests their technology could be used in across wearable electronics, dynamic displays, and even medical devices. The authors do recognize that there are current limitations in their tech and approach. Miniaturization of the components and improved energy efficiency were identified in the research as gaps that will need to be addressed (Yu C. , 2014).

The flexible and “stretchable” design of Yu 's system could be adapted for use on military systems, if they succeed in addressing some of the aforementioned limitations. The response time and ability to replicate complex patterns could enhance VANISH's camouflage capabilities. The combination of both visual and thermal

camouflage in the team's research is also very interesting and could transform VANISH into a more comprehensive concealment solution.

### 2.4 Adaptive Camouflage in Defense: Science Behind Real-Time Color-Changing Tech

The recent progress in adaptive camouflage technology marks an important development milestone for defense-related strategies, according to the 2023 article "Adaptive Camouflage in Defense: Science Behind Real-Time Color-Changing Tech". According to the article, technology that combines sensors, algorithms, and specialized materials to allow real-time blending with surrounding environments is on the way (Businessner, 2023). What makes this tech possible is harnessing the interplay between sensory input, algorithmic processing, and material response. Sensors acting as the eyes of the architecture feed the 'nervous system' algorithms to determine the optimal camouflage configuration for the smart skins. This information is used to manipulate materials embedded with responsive microstructures or pigments, allowing for rapid changes in reflective properties (Businessner, 2023). When combined, these technologies could help military equipment and or personnel achieve unprecedented levels of invisibility (see Figure 7). This advancement may

*Figure 7: Adaptive Camouflage Textiles for NextGen Tanks (Businessner, 2023)*

significantly transform stealth capabilities across different defense systems and platforms.

Adaptive camouflage has many defense applications across military equipment and gear. Vehicles, aircraft, drones, and soldier equipment can all benefit from this tech as it improves stealth and mission accomplishment for tactical, operational, and strategic assets. For example, tanks can change their appearance to match different features, while surveillance drones can better blend with aerial landscapes (Businessner Editorial Team, 2023). The adaptation of this technology for individual soldier equipment should substantially improve combat protection, providing adjustable camouflage that responds to changing surroundings during field operations.

### 2.5 Summary and Conclusion

Despite showing great promise and a realized need on the battlefield, adaptive technology still faces many significant hurdles before it can be widely used by modern defense forces. The systems currently being evaluated in both the civilian and military forums require significant power, ruggedization for challenging environments, and lower price points before they can be meaningfully developed and deploy. Researchers are currently finding inspiration drawn from nature, particularly the camouflage mechanisms of cephalopods, in developing these future technologies that may one day find their way onto the battlefield. This biomimetic approach has led to the creation of effective and versatile systems, particularly in aerospace, which demonstrates the value of interdisciplinary research in advancing military technology (Businessner, 2023). As research in this field progresses and the future of active camouflage solutions become more realized, the battlefield of the 22$^{nd}$ century will undoubtedly transform as well as the soldiers and machines that fight on it. Literature

reviews continue in sections 3.2 and 3.3 of the VANISH Praxis as research delves deeper into the AI/ML algorithms and smart skin materials that may help to realize adaptive camouflage on the battlefield.

# Chapter 3—Methodology

## 3.1 Introduction

The development of the VANISH AI/ML architecture requires a multidisciplinary approach that combines AI/ML applications, materials science, and systems integration. The system design integrates several key components: a flexible, adaptive outer layer capable of changing color and texture; an EO sensor to detect environmental conditions; an onboard processor to analyze sensor data for real-time decision-making and an AI/ML codebase to cast the appropriate adaptive camouflage technique onto the smart skin (Fi 27 for a graphical representation of this framework).

To properly integrate and assess all these subcomponents, the VANISH Praxis methodology devised three key stages to help guide the research:

1. *AI/ML Algorithm and Baseline Code Development Stage*: This stage is where the research and baseline code development occurs for the AI/ML algorithms that will produce the necessary real-world camouflage patterns.
2. *Materials Science Evaluation Stage:* This is where the research explores smart skin materials for feasibility, flexibility, and color ranges.
3. *Simulation, Lab, and Field Testing Stage*: These are the final stages that witness the integration of all sub-components into a testing configuration capable of employment in various environments.

## 3.2 AI/ML Algorithm and Baseline Code Development Stage

The focus of this portion of the Praxis is dedicated to answering Research Question 1 and the corresponding Hypothesis. Throughout this phase of the research, several AI/ML algorithms were researched and assessed for their potential validity in both generating and detecting camouflage solutions on the modern battlefield. At the

core of this architecture are techniques that play a crucial role in processing and analyzing visual data from across the visible spectrum. As seen in many commercial applications, AI/ML is well-suited for this task given its ability to recognize spatial hierarchies and image patterns that enable systems to identify and categorize their surrounding terrain and objects.

### 3.2.1 Applied AI/ML Algorithms for Achieving Adaptive Camouflage

In order to achieve the effects of a real-time adaptive camouflage solution for the battlefield, the VANISH codebase needed to evaluate a number of AI/ML algorithms for their effectiveness and ability to process information and reproduce those visual inputs rapidly into a camouflage output. This section will explore in further detail the underlying mathematical background and potential use cases of these algorithms. Table 4, provides a high-level summary of these techniques and their successful or failed tenure in the baseline VANISH code.

*Table 4: Summary of the Algorithms Explored in VANISH*

| AI/ML Function Name | Purpose | Status In VANISH |
|---|---|---|
| **For Answering Research Question 1 – Part 1** | | |
| **Generative Adversarial Networks (GAN)** | Image generation and data augmentation through TensorFlow/ Keras | **Unsuccessful** – Repeated code errors did not generate a proper image output |
| **Convolutional Neural Networks (CNN)** | Image classification, object detection, and computer vision tasks through TensorFlow/ Keras | **Unsuccessful** – Achieved 25% accuracy for image Classification |
| **K-Means Clustering** | Color palette extraction for scene adaptive camouflage through OpenCV library | **Implemented -** Used for MultiCam Technique |
| **Gaussian Blur** | Smooth/blur images for adaptive camouflage through the OpenCV library | **Implemented** – Used for MultiCam and PredCam Techniques |
| **Wavelet Transform** | Image compression and feature extraction from environmental data through the PyWavelet library | **Explored** – Researched but never implemented |

| | | |
|---|---|---|
| **Entropy** | Measures an image's randomness or information content used for VANISH metrics through the OpenCV library | **Implemented** – Used for MultiCam and PredCam Metrics to measure pattern complexity |
| **Gray-Level Co-occurrence Matrix (GLCM)** | Examines the spatial relationship between pixel intensities to analyze an image's texture through scikit-image | **Implemented** – Used in conjunction with OpenCV for texture analysis. |
| **For Answering Research Question 1 – Part 2** | | |
| **Regional Convolutional Neural Networks (R-CNN)** | Combines CNNs for object detection using TensorFlow/ Keras. | **Explored** – Researched but never implemented as part of the answer to RQ1/H1 to assess its use in Camouflage Recognition. |
| **Fourier Transform (FT) (CamoFA)** | Breaks down images into their basic frequency patterns. Uses the NumPy or OpenCV library. | **Explored** – Researched as part of the answer to RQ1/H1 to assess its use in Camouflage Recognition. |
| **Kalman Filtering (KF)** | Track environmental changes and predict future states. Uses the OpenCV library. | **Explored** – Researched as part of the answer to RQ1/H1 to assess its use in Camouflage Recognition. |
| **Principal Component Analysis (PCA)** | Reduce the dimensionality of data while preserving its essential features. Uses the scikit-learn and NumPy libraries. | **Explored** – Researched as part of the answer to RQ1/H1 to assess its use in Camouflage Recognition. |
| **You Only Look Once (YOLO)** | Aids computer vision tasks by treating object detection as a regression problem | **Explored** – Researched as part of the answer to RQ1/H1 to assess its use in Camouflage Recognition. |
| **Mixture of Gaussian (MOG) Model** | Motion detection for adaptive background modeling OpenCV library | **Tested & Explored** – Initially tested as part of the VANISH codebase but replaced with Gaussian Blur. Researched as part of the answer to RQ1/H1 to assess its use in Camo Recognition. |

#### 3.2.1.1 Generative Adversarial Networks (GANs) Algorithm

Generative Adversarial Networks (GANs) use two neural nets working in concert to create a synthetic image. The way it works is that one part of the algorithm

creates fake data while the other tries to spot what is real versus fake. As the GAN runs, it gets better at creating convincing fakes, while the other half of the model works to improve at spotting them (Ng, 2024).

In its mathematical form, Generative Adversarial Networks (GANs) are given by:

$$min_G\ max_D\ V(D,G) = E_xp_data(x)[log\ D(x)] + E_zp_z(z)[log(1 - D(G(z)))]$$

*Equation 1: Generative Adversarial Networks (GANs) Algorithm (Goodfellow I. J., 2014)*

- Where $p_data(x)$ is the real data distribution
- $p_z(z)$ is the prior distribution on input noise variables
- $G(z)$ is the generator mapping from noise z to generated samples
- $D(x)$ is the discriminator's estimate of the probability that x is real

Within the VANISH construct, GANs can be employed for real-time pattern generation and texture synthesis. This will allow VANISH to create unique camouflage patterns and adapt its appearance accordingly. Future AI/ML research scientists could use GANs to generate different types of terrain-matched textures and employ them alongside CNNs for better camouflage techniques and adaptability. In the paper *"Two-Way Generation of High-Resolution EO and SAR Images via Dual Distortion-Adaptive GANs"* researcher Yuanyuan Qing at the School of Electrical and Electronic Engineering, Nanyang Technological University describes his teams efforts on translating between synthetic aperture radar (SAR) and electro-optical (EO) imagery, The researchers identify a critical limitation in existing deep learning translation methods: non-linear geometric distortions between SAR and EO images caused by different imaging principles, which particularly affect high-resolution imagery (see Figure 8). To solve this problem, they introduce a new sub-meter resolution dataset (SN6-SAROPT) and develop a novel algorithm that adaptively handles these distortions. Their approach demonstrates a better performance for both

SAR-to-EO and EO-to-SAR translation tasks, especially in complex urban environments, which has significant implications for GAN-based cross-domain image synthesis that must account for fundamentally different physical changes across a dynamic landscape environment (Qing, 2023).

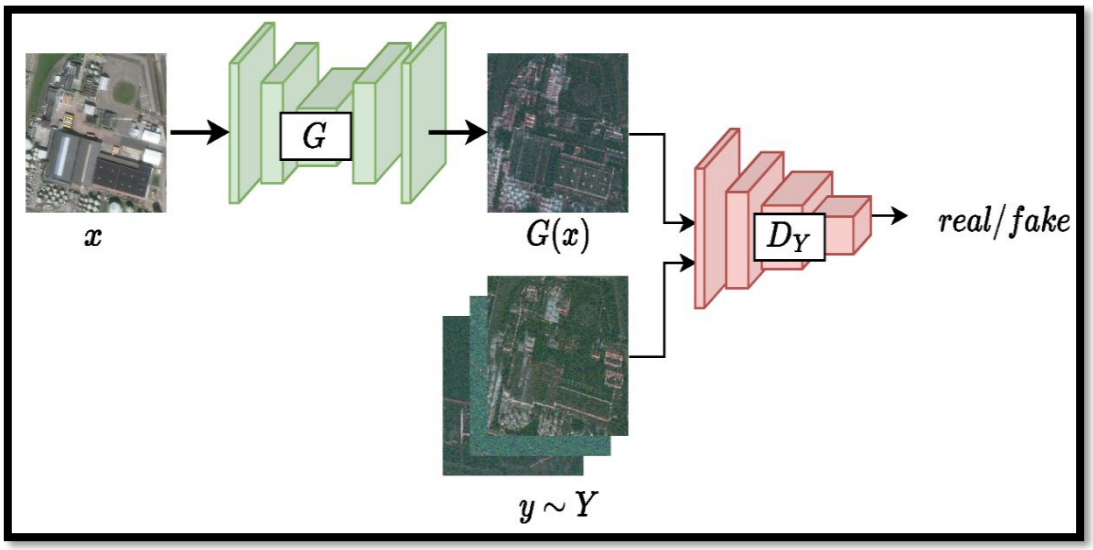



*Figure 8: GANs for SAR to EO Translation SN6-SAROPT Dataset (Qing, 2023)*

**3.2.1.2 Convolutional Neural Networks (CNNs) Algorithm**

Convolutional Neural Networks (CNNs) are deep learning algorithms that work similarly to how human vision processes images (Goodfellow, 2016). This technique is useful in recognizing visual patterns, which makes them well-suited for creating synthetic data. These networks learn from thousands of example images to identify key camouflage elements like textures, colors, and shapes (Goodfellow, 2016). CNNs operate through three main components: convolutional layers that spot patterns using shared filters, pooling layers that compress images while keeping essential details, and fully connected layers that make final determinations. This structure enables CNNs to handle complex visual tasks including image recognition, object detection, and feature analysis (vitalflux.com, 2021).

$$y_l = f(W_l * x_l + b_l)$$

*Equation 2: Convolutional Neural Networks (CNNs) Algorithm (Goodfellow I. , 2016)*

- Where $y_l$ is the output of the layer $l$
- $x_l$ is the input
- $W_l$ is the weight matrix
- $b_l$ is the bias
- $*$ denotes convolution
- $f$ is an activation function

In VANISH it's theorized that a CNN could be trained to recognize different types of terrain (forest, desert, urban, etc.) and automatically select the most appropriate camouflage pattern. In the research paper *"Fast Self-Adaptive Digital Camouflage Design Method Based on Deep Learning"* author Houdi Xiao proposes a fast-self-adaptive digital camouflage design method using deep learning for next-generation adaptive optical camouflage technology. Using CNNs, the research team developed a system architecture that effectively produced background-matched camouflage patterns with strong optical concealment properties (Xiao, 2020).

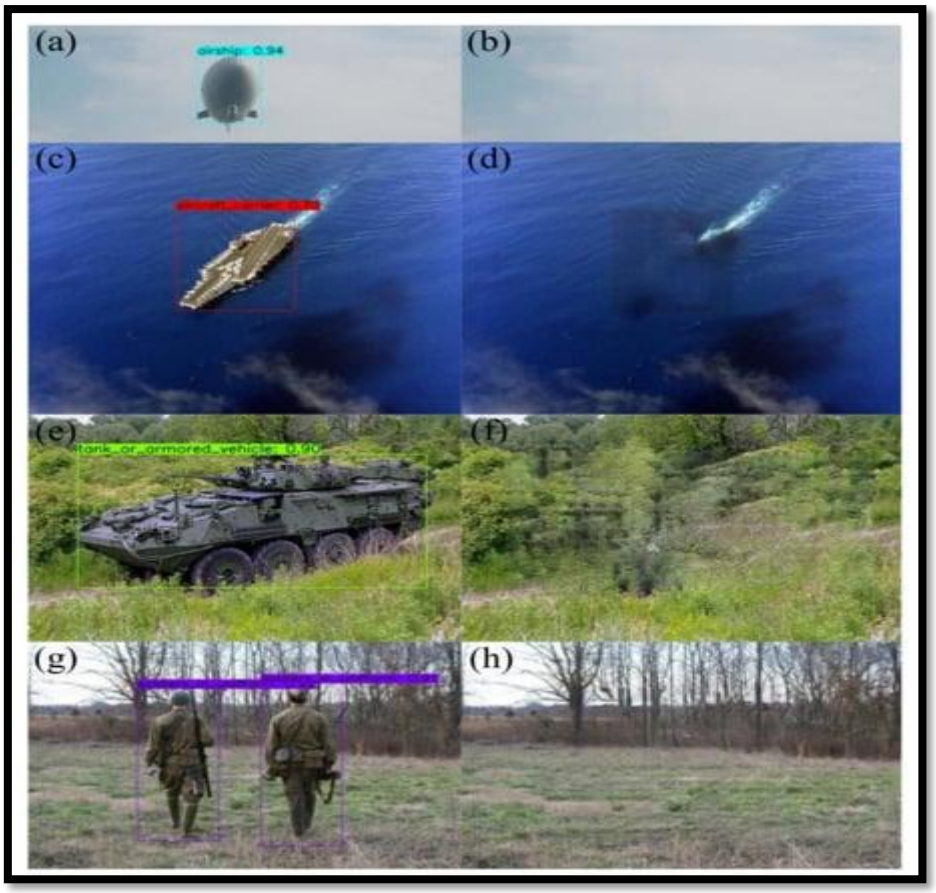



*Figure 9: CNN Applications in Advanced Military Camouflage (Xiao, 2020)*

### 3.2.1.3 K-Means Clustering Algorithm

The K-means clustering algorithm is an unsupervised learning algorithm used for clustering activities. K-means will usually generate K clusters based on the distance between data points and the cluster mean (Zhang, 2025).

In its mathematical form, the K-Means Clustering Algorithm is given by:

$$J = \Sigma(i = 1\ to\ n)\ \Sigma(j = 1\ to\ k)\ w_ij\ ||x_i - \mu_j||^2$$

*Equation 3: K-Means Clustering Algorithm (Wikipedia, 2025)*

- Where x_i = color vector for pixel i (RGB values)
- μ_j = centroid j (cluster center)
- w_ij = 1 if pixel i belongs to cluster j, 0 otherwise
- k = number of color clusters
- n = total number of pixels

In VANISH it's theorized that the K-Means Clustering Algorithm could be used to recognize and generate adaptive camouflage based on real-time EO inputs from various sensors. In the research paper *"Combining Max pooling-Laplacian theory and k-means clustering for novel camouflage pattern design"* authors Minhao Wan, Dehui Zhao, and Baogui Zhao at the Department of Airfield and Construction Engineering, Aviation Engineering School, Air Force Engineering University, Xi'an, China propose using K-Means to create a static digital camouflage pattern design that can handle large image ingestions over a wide field of view (Minhao Wan, 2022).

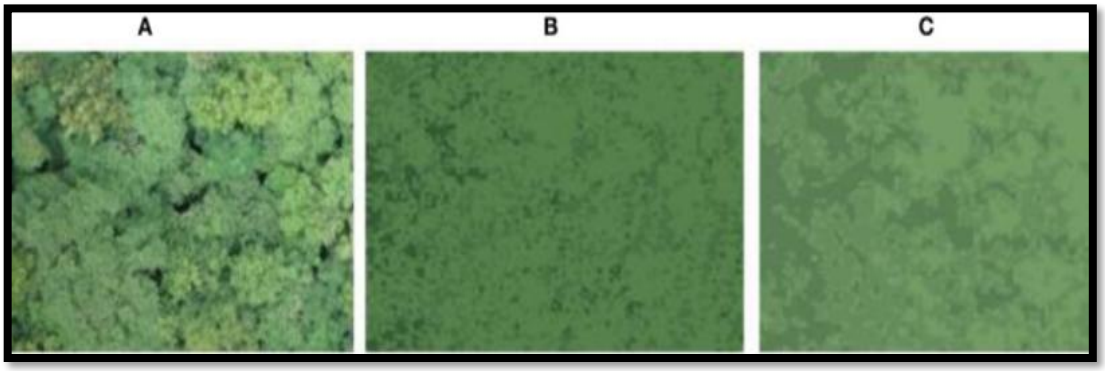



*Figure 10: K-Means for Camouflage Design (Minhao Wan, 2022)*

### 3.2.1.4 Gaussian Blur Algorithm

Founded by Carl Friedrich Gauss, Gaussian Blur is the result of blurring an image by a Gaussian function in order to reduce image noise and detail (Wikipedia, 2025). The effect of this technique is to create a smooth visual blur "resembling that of viewing the image through a translucent screen" (Wikipedia, 2025).

In its mathematical form, the Gaussian Blur Algorithm is given by:

$$G(x, y) = (1/(2\pi\sigma^2)) * e\text{^}(-(x^2 + y^2)/(2\sigma^2))$$

*Equation 4: Gaussian Blur Algorithm (Wikipedia, 2025)*

- Where σ (sigma) is the standard deviation controlling blur strength

- x,y are pixel coordinates relative to the kernel center

In VANISH, the use of Gaussian Blur techniques was proposed and implemented to create the Predator-like effect for a new form of active camouflage called PredCam. This technique then allows the platform to blur into the natural environment, with the goal of evading detection.

*Figure 11: Gaussian Blur Effects on Original Input Stream (Computer Hope, 2024)*

### 3.2.1.5 Information Entropy Formula

Information Entropy, developed in 1948 by Claude Shannon, is a measure of uncertainty, or variability and can be used to measure an image's randomness or information content (Statistics How To, 2025).

In its mathematical form, Entropy is given by:

$$H = \Sigma\, S, i = 1 - (Pi * lnPi)$$

*Equation 5: Entropy Formula (Statistics How To, 2025)*

- Where: H = Shannon Entropy
- Pi = fraction of population composed of a single species i
- ln = natural log
- S = how many species encountered
- Σ = summation of species 1 to S

Within VANISH, Shannon Entropy is used to measure the pattern complexity of the camouflaged images produced in the system. The metrics can later inform future developers and operators of the effectiveness of the chosen camouflage patterns within their area of operations.

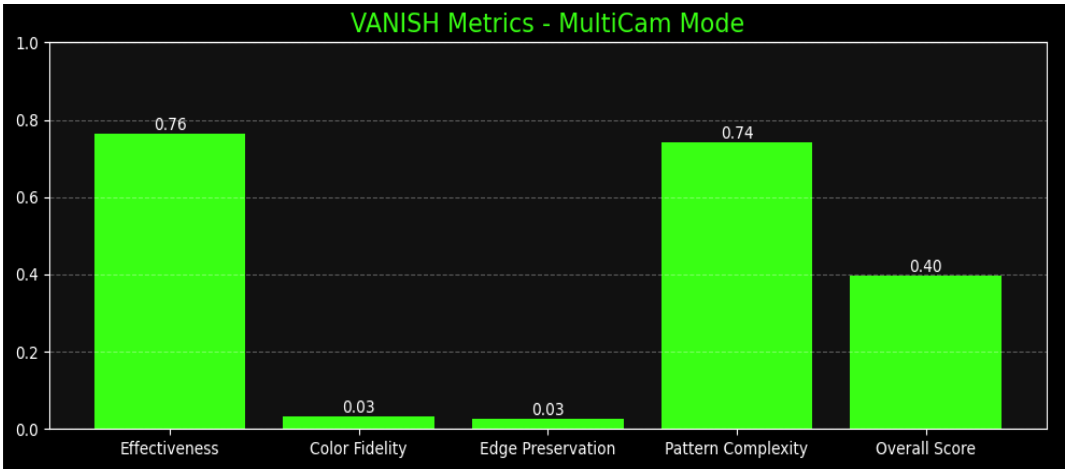



*Figure 12: VANISH Metrics and Applied Entropy to Measure Pattern Complexity*

### 3.2.1.6 Gray-Level Co-occurrence Matrix (GLCM)

Gray-Level Co-occurrence Matrix (GLCM) is a texture classification function from Scikit-Image that delivers a given offset over an image. “Two features of the GLCM matrices are computed: dissimilarity and correlation” (Scikit Image, 2024).

In its mathematical form, GLCM is given by:

skimage.feature.graycomatrix(image, distances, angles, levels=None, symmetric=False, normed=False)

*Equation 6: GLCM Formula (Tutorials Point, 2025)*

- Where: image (array_like): input image data
- distances (array_like): pixel distance offsets
- angles (array_like): pixel pair angles
- levels (int, optional): gray level count
- symmetric (bool, optional): makes matrix symmetric
- normed (bool, optional): normalizes matrix values

Within VANISH, GLCM is used to measure the overall texture of the camouflaged images produced in the system.

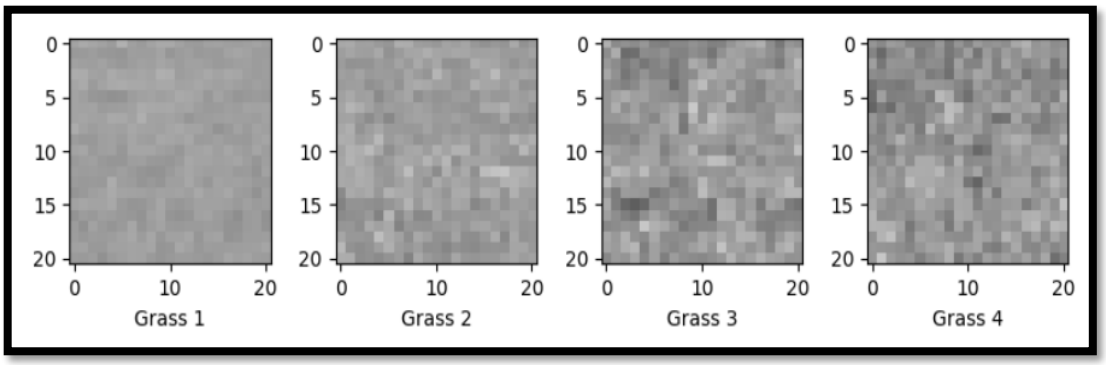



*Figure 13: GLCM for Texture Classification (Scikit Image, 2024)*

### 3.2.2 Applied AI/ML Algorithms for Camouflage Detection

In order to answer the full scope of research question 1, the VANISH Praxis had to explore mathematical tools like the Fourier Transform, Kalman Filtering, and Principal Component Analysis (PCA) and You Only Look Once (YOLO) algorithms for their ability to detect, analyze, and classify potential camouflage technologies on the battlefield. The following sections examine each of these approaches in detail, but they were never intended for integration into the VANISH model. This research was conducted to understand how adversarial "watchers" might use these algorithms to search for and locate VANISH-equipped platforms.

#### *3.2.2.1 Fourier Transform (FT) Formula*

The Fourier Transform is a mathematical tool that breaks down images into their basic frequency patterns. The formula examines how often certain patterns repeat across the frame, which aids in identifying the underlying structure of the image. The technique can be valuable for image filtering, compression, and identifying key features (Lee-Thorp, 2021). In future iterations of VANISH testing,

Fourier Transform analysis can be used to study the frequency patterns found in environments and then replicate those same patterns on VANISH to fool the detectors. This approach may create more effective concealment options and uncover camouflage patterns never before considered.

In its mathematical form, the Fourier Transform (FT) is given by:

$$X[k] = \Sigma(n = 0\ to\ N - 1)x[n] * e^{(-j2\pi kn/N)}$$

*Equation 7: Fourier Transform (FT) Formula (Khare, 2024)*

- Where $X[k]$ is the FT of the sequence $x[n]$.
- $k$ is the index of the frequency component.

In 2023, researchers from the University of Science Ho Chi Minh City explored the concept of Camo Fourier (CamoFA) in their paper "*CamoFA: A Learnable Fourier-based Augmentation for Camouflage Segmentation*" which proposes implementing a learnable augmentation method for camouflaged object detection (COD) and camouflaged instance segmentation (CIS). CamoFA operates in the frequency domain using Fourier Transforms and employs a conditional GAN with cross-attention to generate reference images. It then uses adaptive hybrid swapping to mix low-frequency components from reference images with high-frequency components from input images, making camouflaged objects more discernible (see Figure 14) (Minh-Quan Le, 2023).

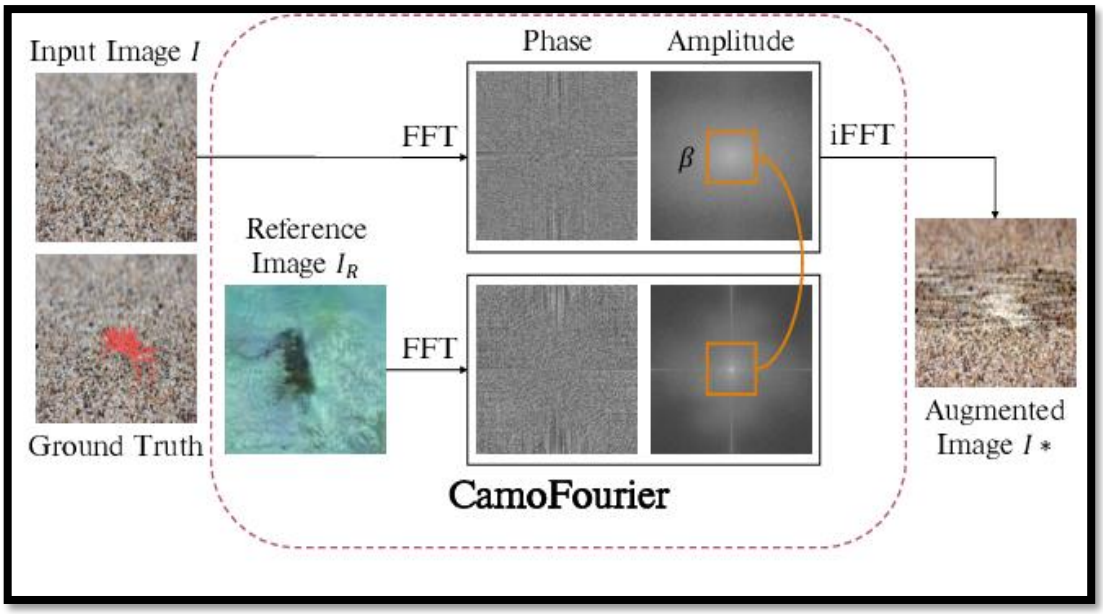



*Figure 14: Fourier Transform in Adaptive Camouflage – CamoFourier (Minh-Quan Le, 2023)*

**3.2.2.2 Kalman Filtering (KF) Formula**

The Kalman Filter is an algorithm that provides estimates of some unknown variables given the measurements observed over time. The Kalman Filter, developed by Rudolf E. Kálmán in the early 1960s, works by continuously updating its predictions as new information comes in. It performs particularly well with systems that follow predictable patterns and have manageable levels of background noise (Kalman, 1960).

Within the construct of VANISH, the filter can improve image processing for applications that may involve fast changing landscapes. The filter was designed to constantly refine predictions by comparing them with actual observations, which helps filter out interference and improve accuracy. For advanced camouflage applications, this means the technology can make real-time adjustments to patterns based on changing surroundings and different viewing angles (Jianqin Yin, 2011). When integrated with VANISH technology, the Kalman Filter could make photonic

smart-skin materials more responsive and effective at avoiding detection across various environmental conditions.

In its mathematical form, the Kalman Filter (KF) is given by:

$$\hat{x}_k = \hat{x}_{\bar{k}} + K_k\left(z_k - H\hat{x}_{\bar{k}}\right)$$

*Equation 8: Kalman Filtering (KF) Formula (Kalman, 1960)*

- Where $\hat{x}_k$ is representative of the updated estimate
- $\hat{x}_{\bar{k}}$ is the predicted state estimate
- $K_k$ is the Kalman gain
- $z_k$ is the measurement
- $H$ is the measurement model

In the context of VANISH concealment, Kalman Filtering could be employed to track environmental changes and predict future states. In the research paper "*Detection of the Mobile Object with Camouflage Color Under Dynamic Background Based on Optical Flow*" by Jianqin Yin the researchers propose a method for detecting camouflaged moving objects against dynamic backgrounds using optical flow modeling and clustering (Jianqin Yin, 2011). After identifying potential target regions through optical flow magnitude histogram analysis and Euclidean distance-based clustering, Kalman Filters are applied to smooth the detection results by filtering noise from the object's location and size parameters using a linear motion model (see Figure 15).

*Figure 15: Results of Object Detection and Optical Flow Using Kalman Filters (Jianqin Yin, 2011)*

### 3.2.2.4 Principal Component Analysis (PCA) Algorithm

Principal Component Analysis is a technique used to reduce the dimensionality of data while preserving its essential features. The principal components are found by solving the eigenvalue problem (Jolliffe, 2002):

In its mathematical form, the Principal Component Analysis (PCA) is given by:

$$(S - \lambda I)v = 0$$

*Equation 9: Principal Component Analysis (PCA) Algorithm (Jolliffe, 2002)*

- Where S is the covariance matrix of the data
- λ are the eigenvalues
- v are the eigenvectors.

In future VANISH applications, PCA can be used to process hyperspectral image data, allowing the system to identify the most important spectral bands and classify them for camouflage matching (see Figure 16). This could enable VANISH to create effective camouflage across multiple spectral ranges (visible, near-infrared, thermal) without processing all spectral information in real-time. This can lead to advanced

concealment techniques by understanding how feature classification algorithms work to identify variations in the landscape across the spectrum.

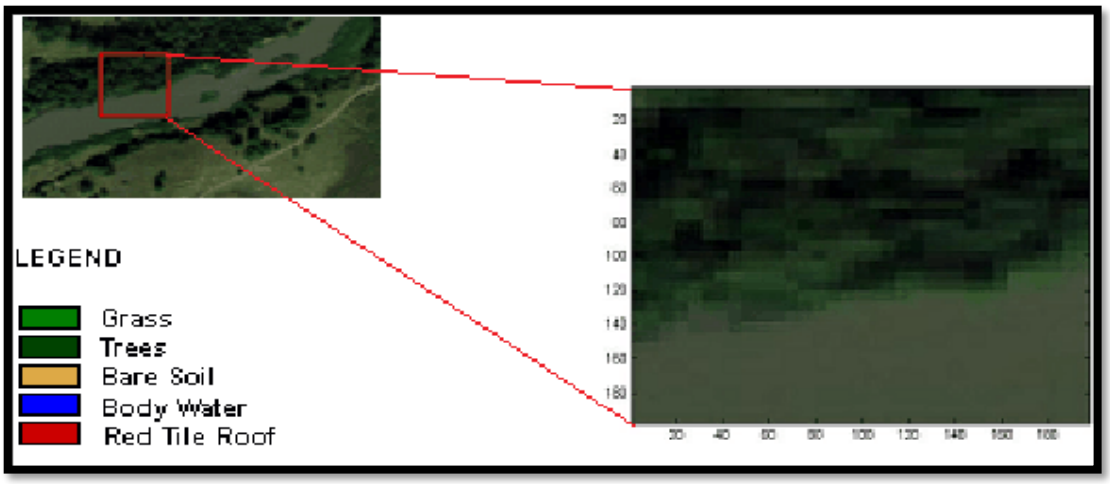



*Figure 16: Principal Component Analysis for Feature Classification (SemanticScholar, 2025)*

### 3.2.2.5 You Only Look Once (YOLO) Algorithm

You Only Look Once (YOLO) is a real-time object detection system that aids computer vision tasks by treating object detection as a regression problem (Redmon J. , 2016). Unlike traditional methods, YOLO divides the image into a grid and predicts bounding boxes and class probabilities for each grid cell in a single forward pass through a neural network.

The core idea can be represented mathematically as:

$$P(Object) * IOU(truth|pred) = P(Class1|Object) * P(Class2|Object) * \ldots * P(ClassN|Object)$$

*Equation 10: You Only Look Once (YOLO) Algorithm (Redmon J. , 2016)*

- Where $P(Object)$ is the probability of an object being present,
- $IOU$ is the intersection over the union of the predicted and true bounding boxes,
- $P(Classi|Object)$ are the conditional class probabilities.

This approach allows YOLO to achieve high accuracy while maintaining real-time performance, processing images at 45 frames per second (for the original version) or

even faster for later versions (Redmon J. , 2016). Future iterations of VANISH could use YOLO to train their models to more rapidly analyze the surrounding environment, detecting and classifying objects, terrains, and textures in real-time. This capability would enable the adaptive camouflage system to quickly identify key elements in its surroundings (see Figure 17) and adjust its appearance for near-instantaneous camouflage adaptation.

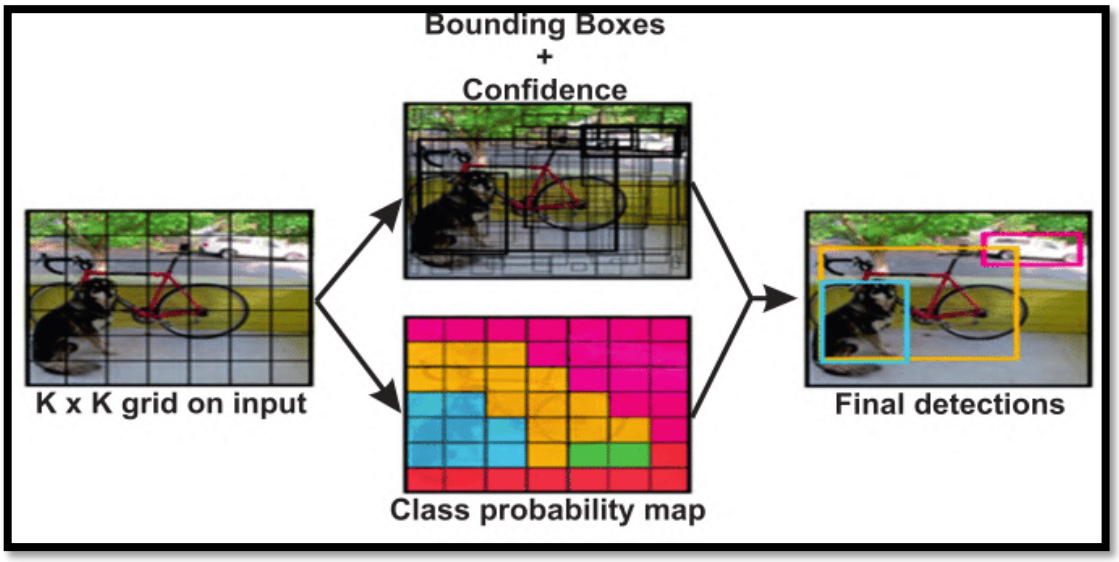



*Figure 17: YOLO Bounding Boxes and Confidence for Class Probability (Joseph, 2025)*

### 3.2.2.6 Mixture of Gaussian (MOG2) Model

The Mixture of Gaussian model is a background subtraction model that is used in some computer vision applications, especially for live streaming video. The value of this system lies in its ability to learn the background of a video and create a stored "memory" of what each pixel normally looks like. When or if a pixel changes significantly from the stored background pattern, it identifies that area as a moving object or foreground element (Tourani, 2020).

The core function can be coded as:

*BackgroundSubtractorMOG2 mog2 = new BackgroundSubtractorMOG2(history, varThreshold, detectShadows)*

*Equation 11: MOG2 Background/Foreground Segmentation Algorithm (Tourani, 2020)*

VANISH, and future testers of the capability, could use MOG2 better map and understand the geographic backgrounds that they are working to blend into. This capability would enable the adaptive camouflage system to quickly identify key elements in its surroundings (see Figure 18) and make adjustments to the camouflage for better scene renderings.

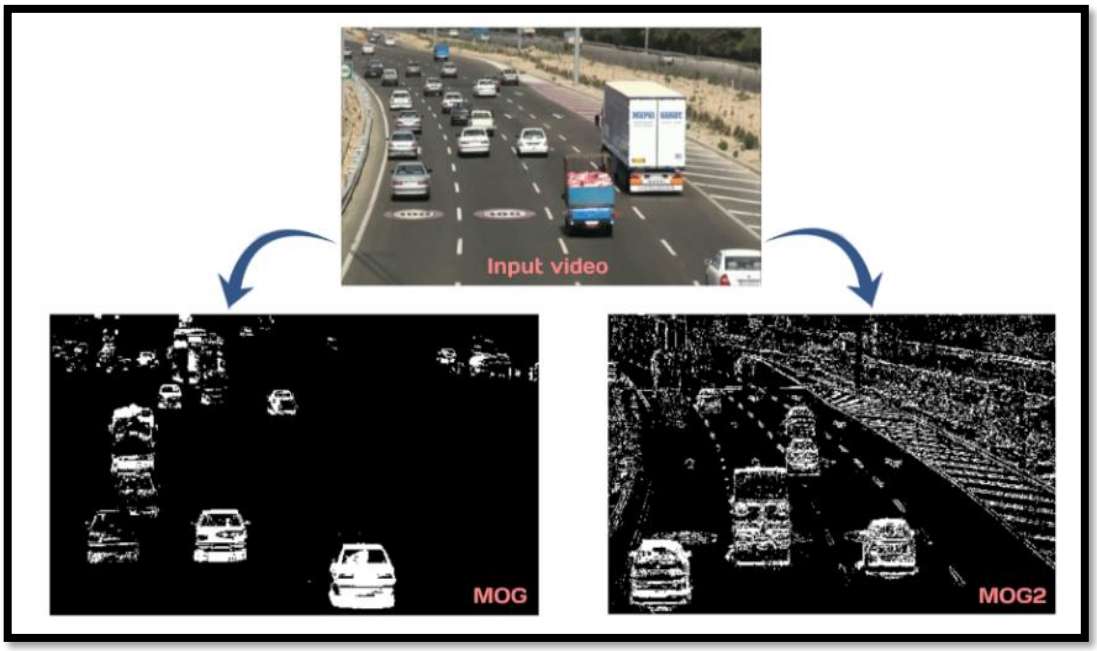



*Figure 18: MOG2 For Background Detection* (Tourani, 2020)

### 3.3 Materials Science Evaluation Stage

This portion of the Praxis is dedicated to answering Research Question 2 and the corresponding Hypothesis. As mentioned previously, VANISH aims to merge advanced AI/ML algorithms with the latest in materials science to create adaptive camouflage for military use. The system incorporates both hardware and software applications to produce camouflage that responds and adjusts to various environmental conditions.

The researched materials all demonstrated potential as foundational elements for camouflage technology. However, it should be noted upfront that each option presents specific limitations regarding durability, activation speed, color spectrum capabilities, and production at larger scales. Further research and development across these advanced materials will be needed to realize a fully capable smart skin solution.

### 3.3.1 Photochromic Materials – Spiropyran

Spiropyran is a compound that changes color when exposed to different wavelengths of light, making it a promising material for VANISH technology. Researchers have been able to demonstrate that these compounds can switch their colors back and forth between different states, which would be useful in adaptive camouflage applications.

#### 3.3.1.1 Chemical Structure and Properties

Spiropyran molecules have a "closed" form that is colorless and transparent, similar to clear glass. When exposed to ultraviolet (UV) these molecules “open” and display a predetermined color pattern (Kortekaas, 2019). The material's ability to change color has interested military researchers who see its potential for improved camouflage. Scientists have already managed to integrate Spiropyran into fabrics that can change color in response to both light and temperature (Abdollahi, 2018). These "smart

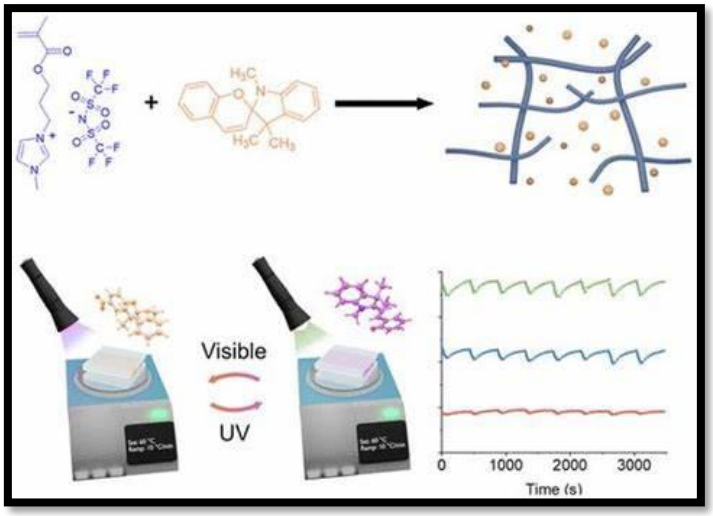


*Figure 19: Spiropyran Molecular Makeup (Rad, 2022)*

fabrics" could potentially allow military vehicles or uniforms to automatically adapt their coloration based on environmental conditions and external stimuli (see Figure 19).

#### 3.3.1.2 Synthesis Methods

According to research conducted by Jaber Rad and his team, the synthesis of Spiropyran-based materials primarily follows for two main approaches: physical doping and covalent anchoring. Physical doping is a method where photochromic dye is directly incorporated into polymer substrates, allowing for higher mobility and sensitivity, but suffering from potential photobleaching and color inhomogeneity (Rad, 2022). Covalent anchoring provides photostability and prevents dye aggregation through the chemical bonding of photoactive groups to polymer chains (Rad, 2022). In simple terms, physical doping is mixing light-sensitive dyes directly into polymers for better responsiveness but risks fading, while covalent anchoring chemically bonds these dyes to the polymer chains for greater stability and even coloration.

### 3.3.1.3 Photochromic Properties

As addressed previously, spiropyrans are color-changing materials that researchers have compared to a switch, turning from clear to colored when exposed to light through "*optically reversible switching membranes*" (Rad, 2022). The photochromic properties of these materials depend on four main things: how much their color changes, how efficiently they use light to make this change, how quickly they can switch back and forth, and how many times they can change before wearing out (Rad, 2022). The ability to manage these performance properties will determine how well these materials work in real-world applications, like camouflage systems that need to change color quickly and reliably.

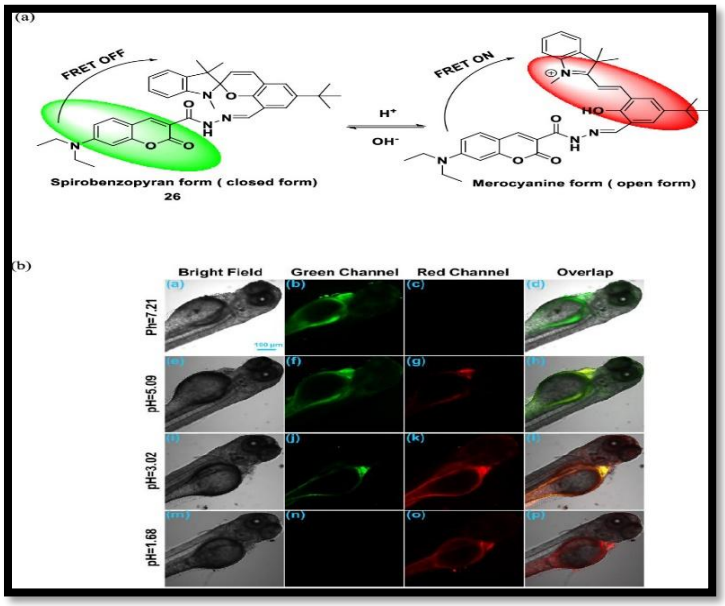



*Figure 20: Spiropyran Color Changing Abilities (Mandal, 2022)*

Mandal's 2022 research article "*Spiropyran–Merocyanine Based Photochromic Fluorescent Probes: Design, Synthesis, and Applications*" explores these color-changing applications in further detail, showcasing how the closed and open forms of the spiropyrans can be used to display varying colors in the Red, Green, Blue (RGB) spectrum (see Figure 20) (Mandal, 2022).

### 3.3.1.4 Challenges and Limitations

Widespread adoption of spiropyran into adaptive camouflage systems like VANISH faces several key challenges that current research aims to address. These

include enhancing the materials to allow for more frequent color changes, expanding the range of light wavelengths that trigger the photochromic response, and integrating spiropyrans with other smart materials for multi-stimuli responsiveness (Kortekaas, 2019). Future research is examining how spiropyran might respond to multiple stimuli at once such as light, temperature, and pressure.

#### 3.3.1.5 Potential for VANISH

The properties discussed previously, make spiropyrans particularly suitable for VANISH applications, where rapid, reversible, and durable color changes are essential. Challenges such as enhancing fatigue resistance, improving thermal stability, and expanding the range of responsive light wavelengths should be addressed before consideration for widespread defense-systems implementation. Current research focuses on developing multi-stimuli responsive spiropyran-based materials that simultaneously adapt to light, temperature, and mechanical stress. This multi-responsive approach could create more versatile and robust adaptive camouflage systems, driving future adaptability for systems like VANISH.

### 3.3.2 Liquid Crystal-Based Smart Skins (LC)

Flexible liquid crystals represent advanced materials capable of altering their appearance and properties. These materials use specialized molecules that self-organize into precise structural arrangements, to create color and sometimes texture changing surfaces. By adding light-sensitive molecules and polymers (long chains of molecules), scientists have created surfaces that can change their color, become

transparent or opaque, and even alter their texture when exposed to triggers like light or heat (see Figure 21) (Bisoyi H. K., 2021). While there are still some challenges to overcome such as making them work well at different temperatures and respond faster these smart skins have promising uses in various military applications.

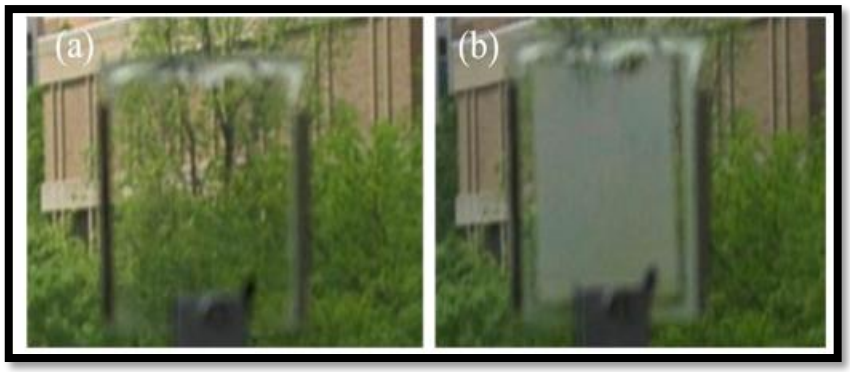



*Figure 21: Real Applications of LCs (Bisoyi H. K., 2021)*

### 3.3.2.1 Chemical Structure and Properties

Liquid crystals occupy a middle ground between regular liquids and solid crystals. These materials are built from rod-shaped or disk-shaped molecules that have a rigid center with flexible ends. For smart skin applications, researchers typically use “thermotropic liquid crystals” because they can reorganize themselves when exposed to different temperatures or other external changes (Bisoyi H. K., 2021).

The most practical type for these applications are “nematic liquid crystals”. These materials have molecules that all point in roughly the same direction, but they're not locked into fixed positions (Bisoyi H. K., 2021). Think of it like a stadium crowd where everyone faces the field to watch the game, but people can still move around and aren't assigned to specific seats. This blend of organization and mobility allows nematic liquid crystals to create smart skins that can adapt their characteristics when needed (Bisoyi H. K., 2021).

### 3.3.2.2 Synthesis and Functionalization

Researchers typically mix these liquid crystals with polymer networks to form what are called “liquid crystal elastomers or networks” (Bisoyi H. K., 2016). This process blends the liquid crystals with existing polymers to produce materials that can respond to environmental changes. In the coming years, these methods could lead to artificial skin that sense and react to its surroundings much like human skin does. The technology builds on established research from recent years that has refined these synthetic approaches (Bisoyi H. K., 2016).

### 3.3.2.3 Photochromic Properties

Like other light-responsive materials previously covered, liquid crystals can change color when exposed to specific types of light. This works by embedding special molecules called spiropyrans into the liquid crystal structure. When light triggers these molecules, the material can shift its color or how see-through it is (Bisoyi H. K., 2021). This makes liquid crystal systems particularly useful for adaptive camouflage that responds to changing light conditions, as well as smart displays and transition technologies that need to adjust their appearance automatically (see Figure 22).

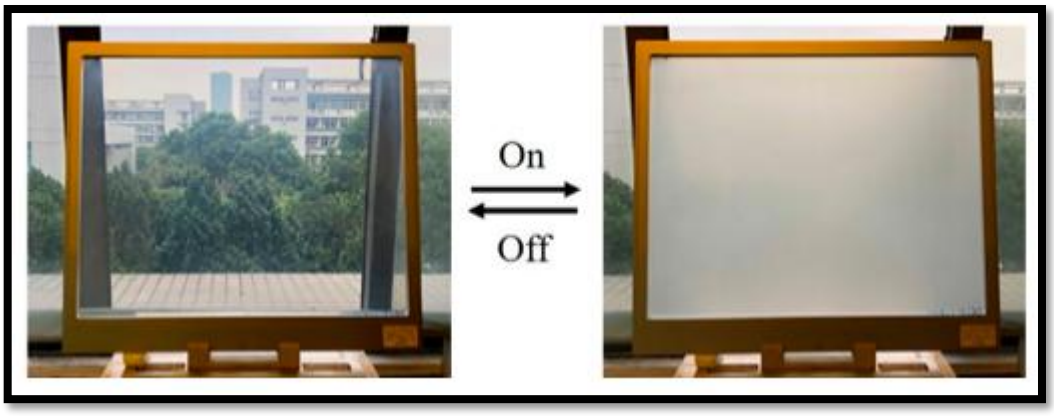



*Figure 22: LCs in Applied Window Transitions* (Bisoyi H. K., 2021)

#### 3.3.2.4 Challenges and Limitations

Several challenges prevent liquid crystals from being practical in military camouflage settings. Temperature sensitivity is a major issue as materials respond much slower when temperature shifts occur, from hot to cold, which could leave equipment visible. Studies from 2016 and 2021 have indicated that these materials can degrade over time, reducing the quality and effectiveness of their visual projections (Bisoyi H. K., 2021). These technical limitations must be resolved before the technology can be deployed in real-world military operations.

#### 3.3.2.5 Potential for Use in VANISH

VANISH could rely heavily on liquid crystals, which can change their optical properties when triggered by external signals. These materials have already proven successful in various commercial fields, from flexible displays to advanced sensor arrays. However, making them work effectively in military applications requires solving several key challenges. The materials need to respond faster, survive harsh combat conditions, and be manufactured at the scales needed for widespread military use. Addressing these technical hurdles will determine whether liquid crystal based VANISH ecosystems can move from laboratory prototypes to practical battlefield applications.

### 3.3.3 Quantum Dot Display (QDD) Smart Skins

Quantum Dot Displays (QDDs) use tiny semiconductor particles to create precise colors through a physics principle called quantum confinement. When these particles are embedded into flexible materials, they can change color in response to different types of light. However, these materials face significant hurdles including toxicity concerns, durability issues, and difficulties in mass production that must be

resolved before they can be widely used. Despite these challenges, their ability to produce highly accurate colors and efficient light output makes them promising for advanced display technology and adaptive surfaces.

#### 3.3.3.1 Chemical Structure and Properties

As discussed above, QDDs have several key advantages such as their color range, ability to absorb light across a wide spectrum, and they don't fade easily under prolonged use. The core material can be comprised of cadmium selenide, indium phosphide, or lead sulfide (Agarwal, 2023). It's this chemical makeup that will allow manufacturers to fine-tune the technology for different applications. This flexibility makes QDD technology particularly valuable for various display needs across a wide range of military consumers.

Note: The lead sulfide component is the lead contributor for toxicity in the display.

#### 3.3.3.2 Synthesis and Functionalization

According to researcher Agarwal, QDDs for smart skin applications are typically synthesized using colloidal and hydrothermal synthesis methods, which offer precise control over their size, shape, and composition (see Figure 23) (Agarwal, 2023). Following synthesis, QDDs undergo functionalization processes to enable their integration into smart skin devices. These processes include "ligand exchange to improve matrix compatibility, bioconjugation for specific sensing applications, and polymer encapsulation to enhance stability" (Agarwal, 2023).

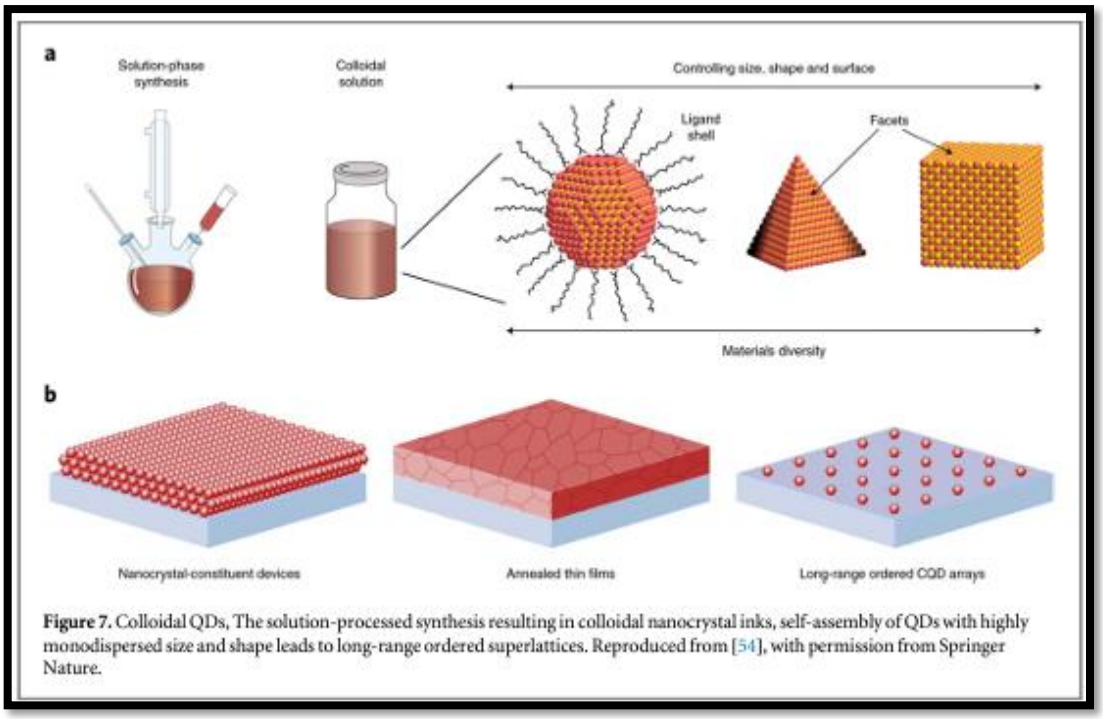



**Figure 7.** Colloidal QDs, The solution-processed synthesis resulting in colloidal nanocrystal inks, self-assembly of QDs with highly monodispersed size and shape leads to long-range ordered superlattices. Reproduced from [54], with permission from Springer Nature.

*Figure 23: Synthesis and Assembly of QDDs* (Agarwal, 2023)

### 3.3.3.3 Photochromic Properties

Key aspects of photochromic QDD smart skins include their ability to achieve reversible color changes, fast response times, high tunability through modifications of QD characteristics, and multi-state switching capabilities (Agarwal, 2023). These features enable the development of advanced display technologies with complex color-changing abilities. Recent research has shown some promising results in the development of hybrid systems that combine QDDs with organic photochromic

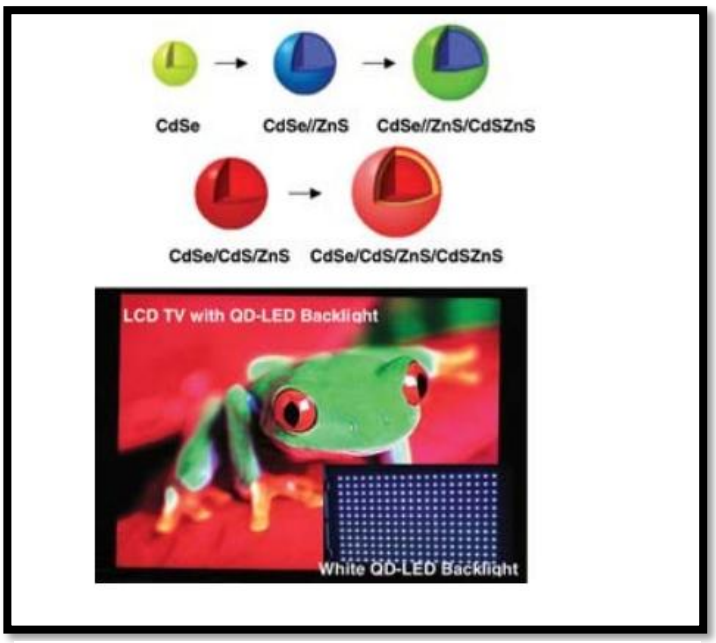



*Figure 24: Quantum Dot Display (Jang, 2010)*

molecules (Jang, 2010). This ability has the potential to lead to enhanced photochromic properties and thereby expand the potential applications of QD-based smart skins in various adaptive camouflage applications (see Figure 24) (Jang, 2010).

#### 3.3.3.4 Challenges and Limitations

Researchers in QDDs are working to address several key limitations to include material durability, broadening the temperature ranges where the technology functions effectively, and improving responsiveness to different light wavelengths. Current efforts have focused on developing materials that can react to multiple environmental factors simultaneously such as light, heat, and pressure (Agarwal, 2023). Resolving these technical challenges would significantly improve the technology's adaptability and performance across varied operational conditions.

#### 3.3.3.5 Potential for Use in VANISH

QDDs are very good at producing exact colors and can change almost instantly when the environment shifts. They're energy-efficient, making them practical for portable military equipment, and they work well in stable weather and lighting conditions. At present price points, the technology is expensive to implement and difficult to manufacture safely, since some quantum materials can be toxic. Ongoing research is assessing how well these displays will hold up under harsh field conditions, and the power systems may need to be redesigned to work optimally with this technology.  VANISH applications may one day implement these materials into their construct, but current scaling limitations are making it difficult to implement and test in a meaningful capacity.

### 3.4 Lab and Field Testing Stage

This stage of the VANISH Praxis combines the best outputs from the previous two stages to develop the adaptive camouflage solution for real-world test employment. Using both data-driven analysis and observational techniques, this phase of testing will showcase for future researchers how the VANISH baseline code works in both lab and field environments. This balanced approach may help ensure the technology works effectively under various conditions.

#### 3.4.1 Experimentation Design

The VANISH ecosystem is a comprehensive process encompassing three primary approaches: simulations, controlled experiments, and field tests.

1. *Simulations*: Comprise of a quick test environment where iterations to the baseline code can be conducted to improve the algorithms in a virtual proving ground.
2. *Controlled Experiments:* Allows for assessing the VANISH code's speed and accuracy at mimicking the environmental surroundings but done in a lab setting.
3. *Field Tests*: These trials are the third and perhaps most important part of data collection. These tests move the technology out of controlled lab settings and into real-world situations, giving better understanding of how the system actually performs under genuine operating conditions.

Figure 25 shows a visual representation of this overall process:

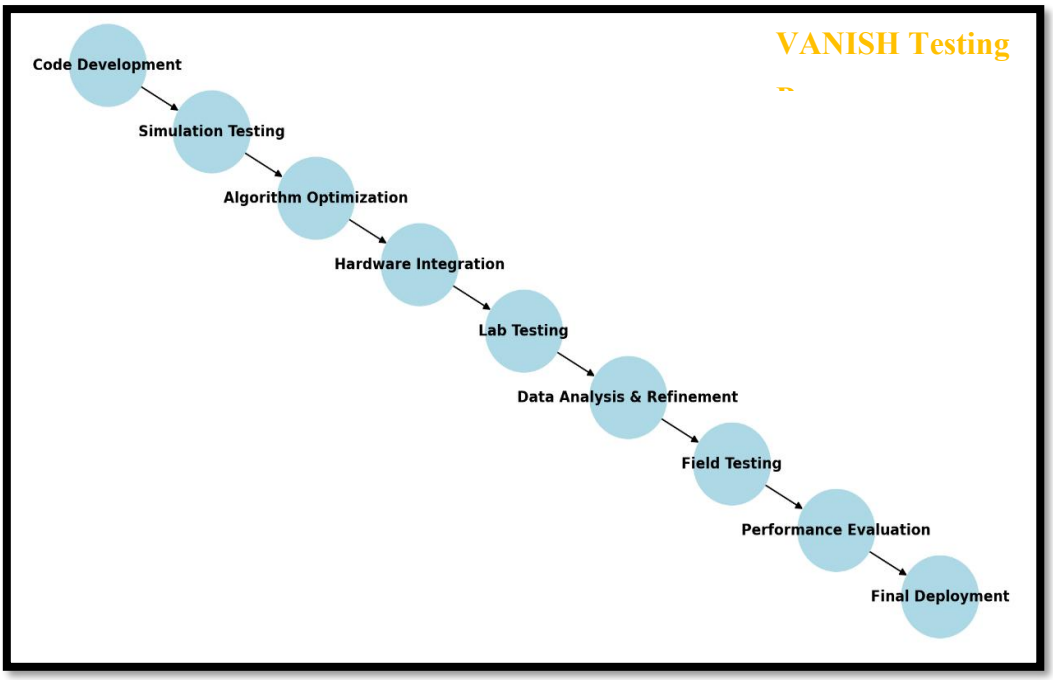



*Figure 25: VANISH Testing Process*

VANISH's testing process phase leverages qualitative and quantitative methods to extract insights from the collected data. Quantitative analysis uses statistics to measure how well the camouflage works at creating accurate patterns and adapting to new surroundings (see Appendix A).

### 3.4.2 Simulation Environment

To develop the VANISH codebase simulation environment, a combination of Integrated Development Environments (IDEs) and platform applications were required to create the machine learning algorithms necessary to achieve adaptive camouflage. Table 4.0 depicts an overview of the critical tools and platforms that were essential in the development of this virtual environment.

*Table 5: VANISH Software Development Tools and Platforms*

| Category | Tool/Platform | Purpose/Features |
|---|---|---|
| **IDEs and Development Environments** | Jupyter Notebook | - Primary Python development<br><br>- Robust debugging tools<br><br>- Version control integration<br><br>- Support for various Python libraries |
| **AI/ML Applications** | TensorFlow, OpenCV, SKLearn | - Computer vision tasks<br><br>- Image processing<br><br>- Real-time environment analysis |
| **AI/ML Application** | NumPy | - Numerical computing and scientific calculations<br><br>- Data processing and analysis |
| **AI/ML Application** | Pandas | - Data manipulation and analysis<br><br>- Handling large datasets |
| **AI/ML Application** | Matplotlib or Plotly | - Data visualization<br><br>- Analyzing camouflage system performance |
| **Compute Hardware** | Raspberry Pi | - Prototyping<br><br>- Potential distributed computing across platforms structure |
| **EO Sensor Hardware** | Sony Camera | - High-speed, low-latency processing of sensor data and video |

### 3.4.3 Flexible LED Smart Skin

VANISH relies heavily on the integration of the aforementioned AI/ML applications into a flexible smart screen which can display the camouflage patterns. After assessing the research conducted in section 3.3 and weighing the options around availability, cost, and functionality, the screen of choice was selected as Light Emitting Diode (LED) produced by a US-based company, HoloscapeLED. The M2 flexible LED array was chosen specifically due to its ability to create a lightweight, flexible, and modular display ( see Figure 26).

*Figure 26: HoloscapeLED M2 Surface Mounted (M2 Panel, 2024)*

### 3.4.4 HoloscapeLED for Operational Evaluation

The HoloscapeLED M2 Surface Mounted Series provides a suitable canvas for creating adaptive camouflage prototypes for VANISH AI/ML code operational demonstrations. While built for indoor advertising, this application could be integrated into a wide range of surfaces, from mobile platforms and buildings to billboards and airline terminal displays (M2 Panel, 2024). This technology offers a wide range of different factory-developed shapes and sizes to quickly meld to the required platform architectures. Table 5.0 depicts the technical specifications for the LED M2 LED array.

*Table 6: HoloscapeLED M2 Specifications (M2 Panel, 2024)*

| Specification | Technical Overview |
|---|---|
| **Product Name** | Holoscape M2 LED Module |
| **Pixel Pitch** | 2.5 mm (horizontal and vertical) |
| **Pixel Density** | 160,000/m² (14,864/ft²) |
| **Apparent Transparency** | 70% |
| **Module Display Dimensions** | M2-1.0m: 125 mm x 1000 mm (4.92" x 39.37")<br>M2-1.2m: 125 mm x 1175 mm (4.92" x 46.26") |
| **Module Profile Dimensions** | M2-1.0m: 125 mm x 1025 mm (4.92" x 40.35")<br>M2-1.2m: 125 mm x 1200 mm (4.92" x 47.25") |
| **Resolution** | M2-1.0m: 50 pixels x 400 pixels<br>M2-1.2m: 50 pixels x 470 pixels |
| **Weight** | 1.63 kg (3.59 lb.) |
| **Brightness** | ≥1200 cd/m² (111.48 cd/ft²) |
| **Scanning Mode** | Static Driving (single pixel, single control) |
| **Encapsulation Type** | Light Board & Driving Board Integrated |
| **Pixel Lifespan** | ≥ 100,000 hours |
| **Pixel Grayscale Depth** | 16 bits |
| **Maximum Power Consumption** | 800 W/m² (74.32 W/ft²) |
| **Average Power Consumption** | 300 W/m² (27.87 W/ft²) |
| **Control System** | Colorlight or Novastar |
| **Input Voltage** | AC100V~240V 50/60Hz |
| **Working Voltage** | DC4.2V ±0.2V |
| **Working Temperature** | -20°C to 50°C |
| **Working Humidity** | Up to 85% RH (without condensation) |
| **Storage Temperature** | -20°C to 85°C |
| **Storage Humidity** | Up to 85% RH (without condensation) |
| **Protection Degree** | IP20 |
| **Installation Environment** | Indoor |

### 3.4.5 Integrated VANISH Design Framework for Simulation, Lab, and Field Testing

Combining the research elements from the aforementioned stages: AI/ML Code Development, Materials Science Evaluation, and the Simulation, Lab, and Testing Stage, the base framework is now established to begin conducting a scientific evaluation of the effectiveness of the VANISH ecosystem.

Figure 27 provides a traceable flow chart that combines the key elements of the VANISH AI/ML architecture and outlines them in a framework.

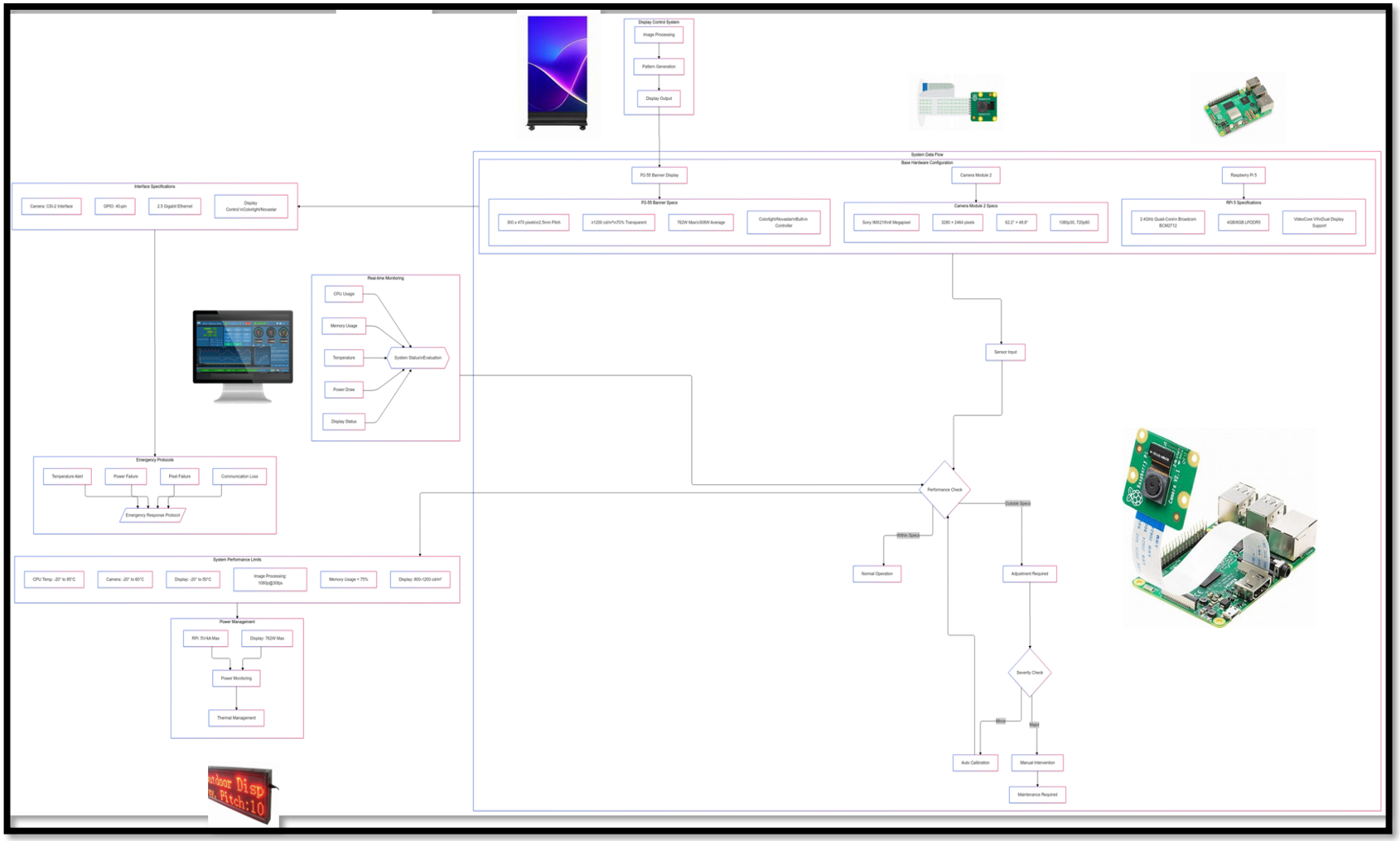



*Figure 27: VANISH System Level Framework*

# Chapter 4 – Results

## 4.1 Introduction

The research stages documented across Chapter 3 helped establish the conceptual framework for the VANISH AI/ML architecture and created the necessary roadmap for achieving the academic outcomes of this Praxis. Implementation of these core concepts proved challenging, requiring a host of starts, stops, and restarts as development and integration of the VANISH ecosystem materialized. Development of the AI/ML codebase for VANISH evolved over the course of multiple developmental phases, beginning with initial CNN classification model training attempts that took place from August to mid-November of 2024. These initial model development efforts were largely unsuccessful in producing an architecture that could deliver real time adaptive camouflage to a smart skin application. The first attempts covered in sections 4.2.2 and 4.2.3 were overly reliant on the input training data and ultimately proved to be the wrong type of models for realizing the Praxis key deliverables.

After abandoning further development efforts across the TensorFlow/Keras libraries, VANISH code development began taking shape while exploring the K-Means Clustering algorithm in the Scikit-Image. Over the next four months these algorithms helped establish the framework for generating MultiCam and PredCam patterns that would become the primary camouflage drivers across the VANISH platform in early 2025.

In parallel with the AI/ML software development efforts, the VANISH Praxis began detailed system integration efforts to establish the necessary connectivity between the baseline code and the LED display panels to display the synthetic

camouflage feeds. Initial VANISH system testing was conducted through controlled laboratory experiments at the HoloscapeLED facility in Orlando, Florida, and followed up in early 2025 with field tests in Apollo Beach, Florida. The field tests exposed the VANISH AI/ML codebase to a wide range of all-domain environments, from maritime and desert conditions to urban and forested settings. These tests were able to assess the system's capabilities across varied lighting conditions, terrain types, and environmental challenges, while uncovering a number of issues with the baseline code, mainly stemming from system latency that ranged from a few seconds to several minutes.

Following the lab and field tests, dramatic improvements to the code were made possible by focusing development efforts on the OpenCV family of libraries. The K-Means Clustering and Gaussian Blur algorithms became the primary drivers of the VANISH baseline found in Appendix A and enabled the real-time streaming of the MultiCam and PredCam applications.

### 4.2 VANISH AI/ML Code Development

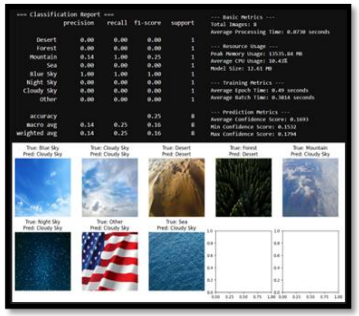
The following sections of the Praxis outline the timeline and key activities that transpired during the 9-month development effort of the VANISH codebase. Each of the six sections has a brief introduction, followed by a Summary Table that outlines the specific models:

- System Level Overview
- Capabilities
- Model Architecture
- Technical Specifications

Each section ends with an image rendering of the model output.

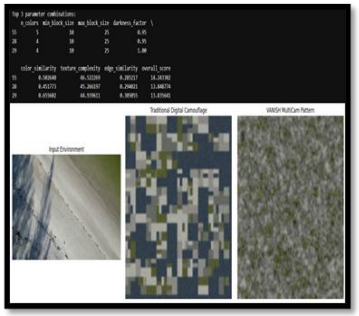
### 4.2.1 VANISH AI/ML Code Development Timeline

**VANISH 1.0**



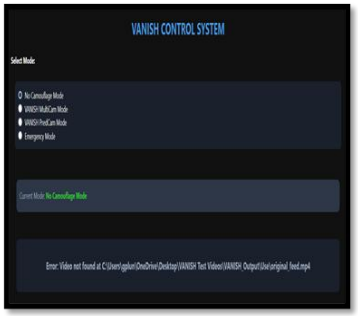
**Outcome:** Unsuccessful
**AI/ML Libraries:** TensorFlow/Keras
**Models Explored:** CNNs and GANs
**Timeframe:** Aug – Nov 2024

**VANISH 3.0**



**Outcome:** Partially Successful – Static Camo Image
**AI/ML Libraries:** OpenCV
**Models Explored:** K-Means Clustering
**Timeframe:** Jan – Feb 2025

**VANISH 5.0**

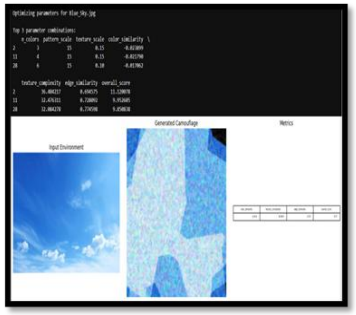



**Outcome:** Partially Successful – Graphical User Interface
**AI/ML Libraries:** OpenCV
**Models Explored:** Gaussian Blur and K-Means
**Timeframe:** Mar 2025

**Timeline**

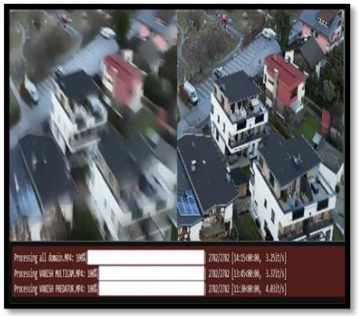
**Outcome:** Unsuccessful
**AI/ML Libraries:** SciKit-Image
**Models Explored:** K-Means Clustering
**Timeframe:** Nov 2024 – Jan 2025

**VANISH 2.0**



**Outcome:** Partially Successful – FMV Feed
**AI/ML Libraries:** OpenCV
**Models Explored:** Gaussian Blur and K-Means
**Timeframe:** Feb – Mar 2025

**VANISH 4.0**



**Outcome:** Partially Successful – Fully Integrated GUI
**AI/ML Libraries:** OpenCV
**Models Explored:** Gaussian Blur and K-Means
**Timeframe:** Apr 2025 - Present

**VANISH 6.0**

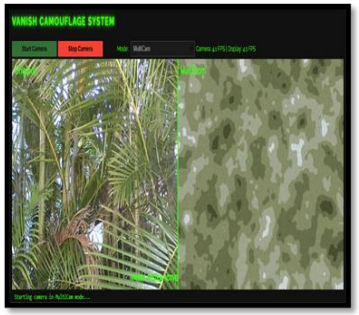



*Figure 28: VANISH Code Development Timeline*

### 4.2.2 VANISH CNN Model Training Attempt

At the early outset of the VANISH Praxis (August-November 2024), it was theorized that a CNN or GAN could be trained to identify various scenes and predict the right correlating response when presented with a number of random images of various environmental pictures. Unfortunately, the introduction of new images into the dataset threw off the predictive analysis and the model would degrade significantly (~25% accuracy scores). Attempts were made to incorporate GANs into the architecture, but the code would fail to process.

**Result:** The other major issue with using CNNs/GANs was that developing an image classifier was not the right path for the VANISH Praxis, since the end goal was to create an adaptive camouflage output to put on smart skin materials. For these reasons, this model was deemed *unsuccessful*, and a model pivot was needed.

*Table 7: VANISH CNN Model Overview*

| **System Overview** | |
|---|---|
| **AI/ML Methods from Chapter 3** | CNNs/GANs via TensorFlow/Keras |
| **Objective** | Scene Classifier for Camouflage |
| **Input Format** | Images (PNG, JPEG, BMP, GIF) |
| **Image Resolution** | 128 x 128 RGB |
| **Capabilities** | |
| **Class Categories** | **Count** |
| **Total Classes** | 8 |
| **Natural Scenes** | Desert, Forest, Mountain, Sea |
| **Sky Types** | Blue Sky, Night Sky, Cloudy Sky |
| **Random Scene** | American Flag |
| **Model Architecture** | |
| **Layer Types** | **Configuration** |
| **Conv2D Layers** | 3 layers (32, 64, 128 filters) |
| **Activation** | ReLU |
| **Normalization** | Batch |
| **Pooling** | MaxPooling2D (2x2) |
| **Dense Layers** | 128 neurons + output layer |
| **Regularization** | Dropout (0.3) |

| | |
|---|---|
| **Optimizer** | Adam |
| **Technical Specifications** | |
| **Spec** | **Value** |
| **Training Split** | 80% train, 20% validation |
| **Epochs** | 10 |
| **Confidence Threshold** | 0.7 (70%) |
| **Batch Processing** | Full dataset |

| | |
|---|---|
| **Overall Status** | Unsuccessful |

Figure 34 depicts a sample output from one of the desktop trials showing the VANISH code with a very low accuracy rate of 25%.

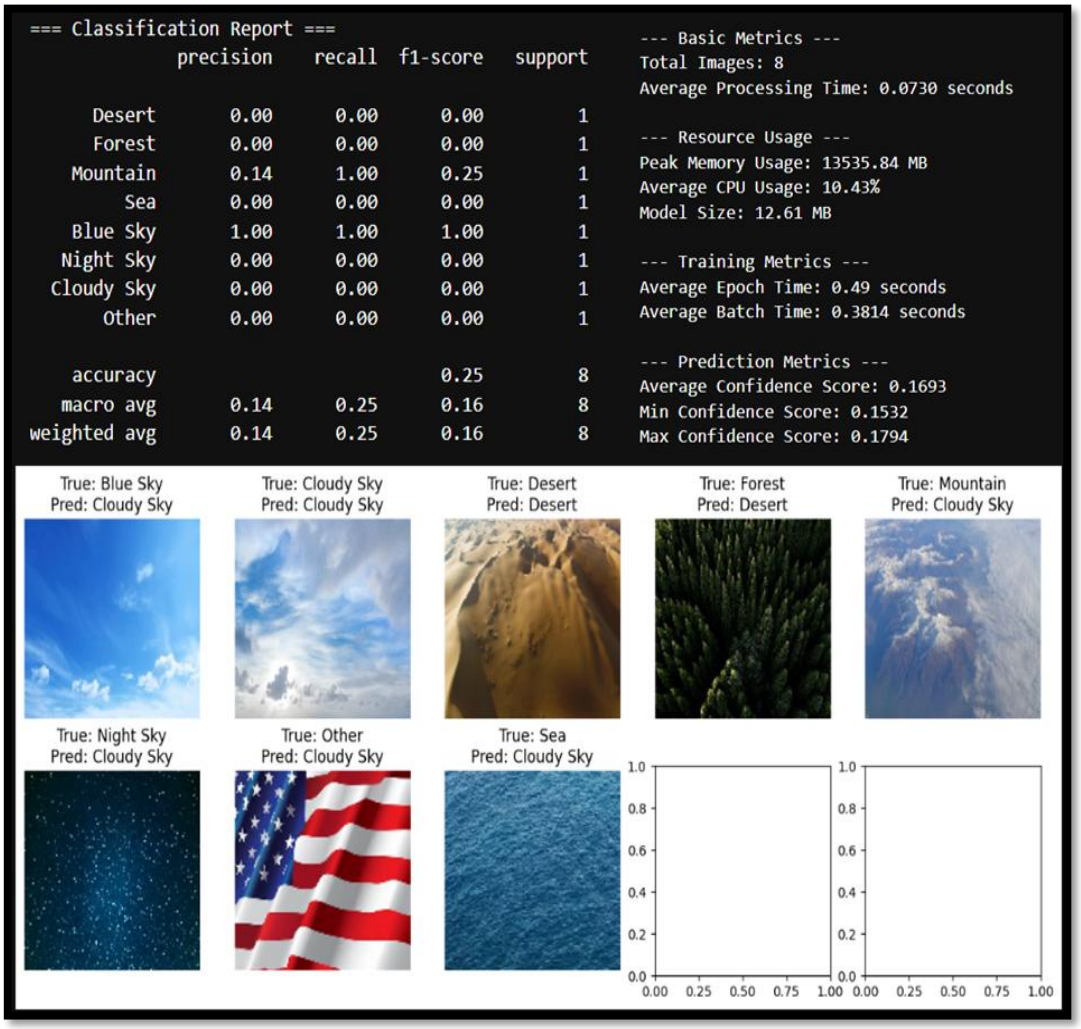



*Figure 29: VANISH Code Attempting to Train a CNN*

### 4.2.3 VANISH Scikit-Learn + SciPy Model for Adaptive Camouflage Attempt

By mid-November 2024, the VANISH code had moved away from attempting to train CNNs for image classification with TensorFlow/Keras and was at the very early phases of exploring K-Means Clustering with the SciKit Learn and SciPy libraries. Initial renderings of output images that displayed digital camouflage variations were encouraging, but streaming these outputs together for real-time display purposes was unrealistic due to latency concerns.

**Result:** The primary reason this code was deemed *unsuccessful* was that the output was just a series of static images, when the end goal was to develop a streaming video output of adaptive camouflage renderings. A computer vision model like OpenCV was likely more appropriate for what VANISH Praxis was trying to achieve.

*Table 8: VANISH Scikit-Learn + SciPy Model Overview*

| **System Overview** | |
|---|---|
| **Description** | **Method** |
| **AI/ML Methods from Chapter 3** | K-Means Clustering via Scikit-Learn + SciPy |
| **Objective** | Camouflage Pattern Generation |
| **Input Format** | Images (JPEG, PNG, BMP) |
| **Image Resolution** | 256 x 256 RGB |
| **Capabilities** | |
| **Stage** | **Method** |
| **Color Extraction** | K-Means Clustering (SKLearn) |
| **Pattern Generation** | Voronoi Diagrams (SciPy) |
| **Texture Application** | Multi-layer Noise |
| **Model Architecture** | |
| **Layer Type** | **Configuration** |
| **K-Means Clustering Algorithm** | Number of Clusters (5) |
| **Color Sorting** | By Brightness |
| **Pattern Scales** | Voronoi (20, 30, 40 points) |
| **Noise Layers** | 3 Values (0.1, 0.05, 0.02) |
| **Technical Specifications** | |
| **Process** | **Output** |

| | |
|---|---|
| **Color Clusters** | N_colors=5 |
| **Pattern Generation** | Pixel clustering |
| **Noise Scales** | Weighted averages of the patterns |

| | |
|---|---|
| **Overall Status** | Unsuccessful |

Figure 35 depicts the output from one of the desktop trials showing the VANISH code produce its first camouflage output when presented with an input picture. Note: The same input image datasets from the first VANISH CNN models were used to test the functionality.

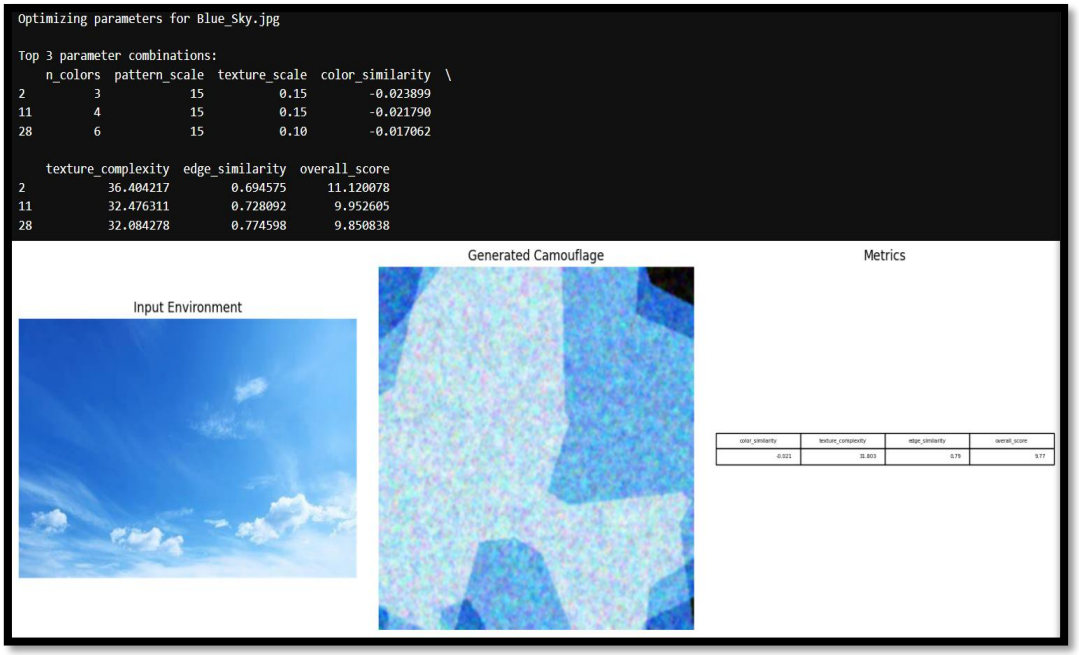



*Figure 30: Generated Camouflage Output*

### 4.2.4 VANISH OpenCV + Scikit-Image Model for Adaptive Camouflage Attempt

In mid-January 2025, at the time of the VANISH field tests (see section 4.4), the third model pivot of the Praxis occurred. During this pivot, the OpenCV library became the primary focus moving forward, using the K-Mean Clustering algorithm (OpenCV) to generate the first static MultiCam camouflage depictions of the research project off a live stream video feed.

**Result:** This code iteration was deemed *partially successful* as it marked the first milestone for the VANISH Praxis, demonstrating its ability to stream live video data into the model and have it render a MultiCam image as the output. Since the output remained a static image, further iterations of the model were needed.

*Table 9: VANISH OpenCV +Scikit-Image Model Overview*

| **System Overview** | |
|---|---|
| **Description** | **Method** |
| **AI/ML Methods from Chapter 3** | K-Means and GLCM via OpenCV + Scikit-Image |
| **Objective** | Develop Adaptable Military Camouflage |
| **Input Format** | Images + Video Files (MP4, JPEG, PNG) |
| **Image Resolution** | 256 x 256 RGB |
| **Capabilities** | |
| **Stage** | **Method** |
| **Digital Camouflage** | Recursive block splitting |
| **MultiCam Pattern** | Multi-layer noise synthesis |
| **Adaptive Coloring** | K-Means color extraction |
| **Video Processing** | Frame-by-frame analysis |
| **Model Architecture** | |
| **Layer Type** | **Configuration** |
| **Color Clustering** | Scikit-Image K-Means + OpenCV |
| **Texture Analysis** | Scikit-Image GLCM |
| **Pattern Generation** | OpenCV |
| **Technical Specifications** | |
| **Process** | **Output** |
| **Color Clusters** | 3 to 5 |
| **Block Size Range** | 10-35 pixels |
| **Darkness Factor** | 0.9-1.0 |
| **Visualization** | Matplotlib + OpenCV |

| | |
|---|---|
| **Overall Status** | Partially Successful |

Figure 31 depicts the output from one of the desktop trials showing the VANISH code produce its first camouflage output when presented with an input video feed.

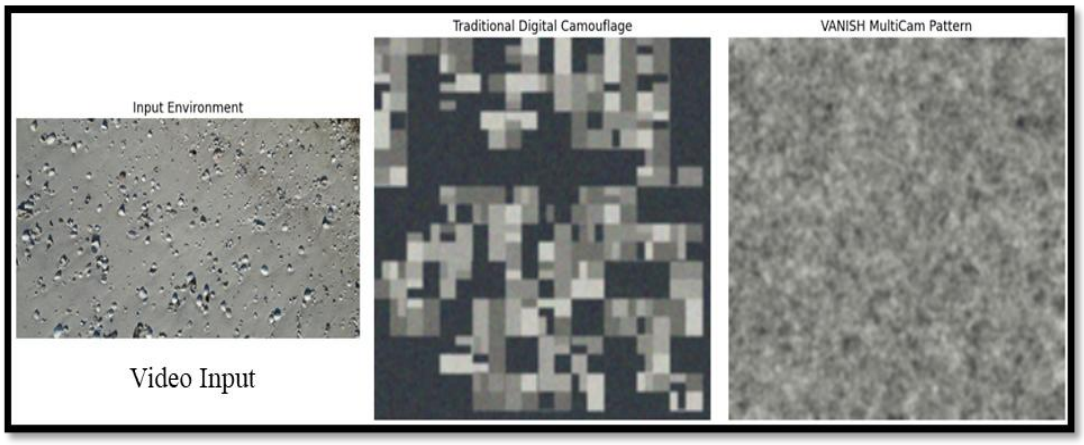



*Figure 31: VANISH OpenCV +Scikit-Image Model First K-Means MultiCam Output*

### 4.2.5 VANISH OpenCV Model for Adaptive Camouflage Attempt

Between February and March 2025, the VANISH codebase was running the OpenCV library for all processing and manipulation of live video inputs. It was at this time that the introduction of Predator Camouflage (PredCam) was introduced through the incorporation of Gaussian Blur as a new and novel camouflage technique that would complement the MultiCam instance that was already developed in the previous code iteration.

**Result:** This code iteration was also deemed *partially successful* as it began to show the diversity of the OpenCV library and the additional capabilities that VANISH could realize. The system was still manually intensive, and a graphical user interface was still required for the system to work properly, therefore further iterations of the codebase were needed.

*Table 10: VANISH OpenCV Model Overview*

| **System Overview** | |
|---|---|
| **Description** | **Method** |
| **AI/ML Methods from Chapter 3** | K-Means, Gaussian Blur, Information Entropy via OpenCV + Scikit-Image |
| **Objective** | Live Stream Adaptive Camouflage |

| | |
|---|---|
| **Input Format** | Video files (MP4, AVI) |
| **Image Resolution** | 256 x 256 |
| **Capabilities** | |
| **Stage** | **Method** |
| **Frame Extraction** | Sequential video frame capture |
| **Metrics Calculation** | Brightness, contrast, standard deviation |
| **Video Reconstruction** | Multi-codec frame compilation |
| **Batch Processing** | Video analysis |
| **Model Architecture** | |
| **Layer Type** | **Configuration** |
| **Video Smoothing (PredCam)** | OpenCV Gaussian Blur |
| **Frame Processing (MultiCam)** | OpenCV K-Means |
| **Metrics Engine** | NumPy statistical analysis |
| **Display System** | OpenCV real-time display |
| **Technical Specifications** | |
| **Process** | **Output** |
| **Supported Types for Video Injects** | H264, XVID, MJPG |
| **Frame Formats** | PNG intermediate files |
| **Resize Target** | 256 x 256 pixels |
| **Export Formats** | CSV metrics, MP4/AVI output |

| | |
|---|---|
| **Overall Status** | Partially Successful |

Figure 32 depicts the VANISH code producing its first live video output of Predator Camouflage (PredCam) while running a live video input feed.

*Figure 32: VANISH OpenCV Model First PredCam Output*

### 4.2.6 VANISH OpenCV Model Initial Graphical User Interface Design and Integration

By late March 2025, the VANISH OpenCV model was running quite well and a graphical user interface (GUI) was required to allow end users the functionality to switch between MultiCam, PredCam and No Camouflage instances at their discretion. The concept also emerged that end users may want to openly telegraph their location to the world in times of emergency, as such an "Emergency" camouflage feature was developed that displayed a blaze orange pattern.

**Results:** This phase of the Praxis still had the overall code achieving a *partially successful* score as initial integrations between the GUI and VANISH codebase failed to display the requested video feeds and several of the GUI radio buttons did not function appropriately or at all. Future iteration of the VANISH code and GUI were required that could ingest a live stream.

*Table 11: VANISH Graphical User Interface (GUI) Model Overview*

| **System Overview** | |
|---|---|
| **Description** | **Method** |
| **AI/ML Methods from Chapter 3** | K-Means, Gaussian Blur, Information Entropy via OpenCV IPyWidgets + HTML/CSS |
| **Objective** | VANISH System User Interface |
| **Input Format** | User interactions (radio buttons) |
| **Output Format** | Live video feeds + status display |
| **Capabilities** | |
| **Stage** | **Method** |
| **No Camouflage Mode** | Original video feed display |
| **VANISH MultiCam Mode** | MultiCam pattern video output |
| **VANISH PredCam Mode** | PredCam pattern video output |
| **Emergency Mode** | High-visibility emergency output (orange) |
| **Model Architecture** | |

| Layer Type | Configuration |
|---|---|
| **GUI Framework** | IPyWidgets |
| **Styling System** | HTML/CSS integration |
| **Video Display** | HTML5 video tags (failed) |
| **Technical Specifications** | |
| **Process** | **Output** |
| **Interface Type** | Jupyter Notebook |
| **Video Formats** | MP4 support |
| **Display Modes** | 4 operational modes |
| **Styling** | Custom CSS Dark Predator Theme |

| | |
|---|---|
| **Overall Status** | Partially Successful |

Figure 33 depicts the output from the first iteration of the VANISH GUI, with the initial integration error that failed to fetch the incoming video feeds.

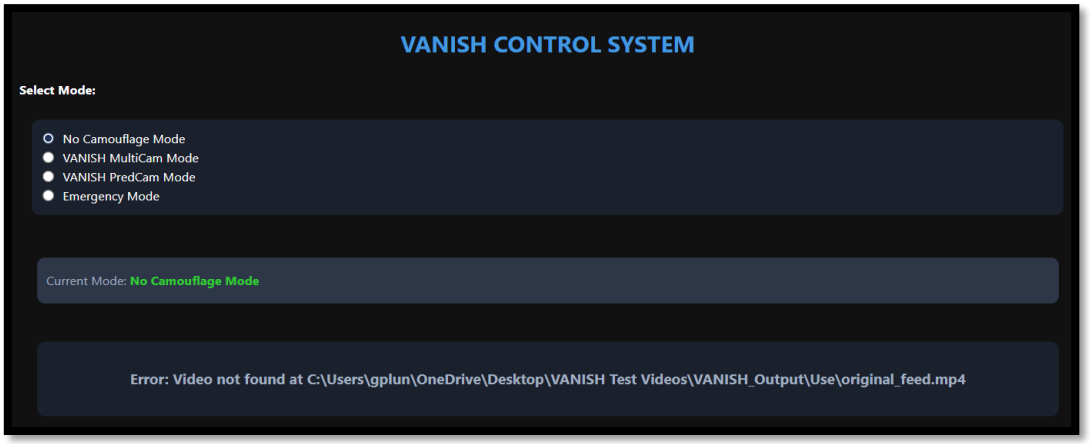



*Figure 33: Initial VANISH GUI Output*

### 4.2.7 VANISH AI/ML Model Final

In mid- April 2025, the final fully integrated version of the VANISH model was completed. This code is provided in its entirety for future researchers to assess and iterate on in Appendix A of this Praxis.

**Result:** The final iteration of VANISH code is deemed *successful* as it is able to incorporate a large number of the concepts from Chapter 3. RQ1 and RQ2 will

only be able to be answered in full once this code is paired with a smart skin and operationally tested.

*Table 12: VANISH AI/ML Model Final Overview*

| **System Overview** | | |
|---|---|---|
| **Description** | | **Method** |
| **AI/ML Methods from Chapter 3** | | K-Means, Gaussian Blur, Information Entropy via OpenCV + IPyWidgets + Matplotlib |
| **Objective** | | VANISH Adaptive Camouflage |
| **Input Format** | | Real-time video stream |
| **Output Format** | | Live Camera Camouflage Effects |
| **Capabilities** | | |
| **Stage** | | **Method** |
| **No Camouflage Mode** | | Original video feed display |
| **VANISH MultiCam Mode** | | Adaptive pattern generation |
| **VANISH PredCam Mode** | | Edge-preserving camouflage |
| **Emergency Mode** | | High-visibility emergency feed |
| **Model Architecture** | | |
| **Layer Type** | | **Configuration** |
| **GUI Framework** | | IPyWidgets |
| **Image Processing** | | OpenCV |
| **Visualization** | | Matplotlib |
| **Metrics** | | Shannon Entropy |
| **User Interface** | | Interactive widgets |
| **Technical Specifications** | | |
| **Process** | | **Output** |
| **MULTICAM** | **Color Clusters** | n_colors = 5 number of colors to extract |
| | **Noise Scales** | noise_scale = 40, 20, 10 for large, medium, and fine pattern features |
| | **Darkness Factor** | darkness_factor = .95 set for tactical appearance |
| | **Final Blur** | (3, 3) small smoothing for natural texture |
| **PREDCAM** | **Spatial Smoothing Radius** | sigma_s = 200 aggressive smoothing |
| | **Color Blending Threshold** | sigma_r = .9 aggressive color blending |
| | **Bilateral Kernal** | 20 (neighborhood size) |
| | **Color Quantization** | 24 - divides 256 into 11 levels per channel |

| | |
|---|---|
| **Overall Status** | Successful |

Figure 34 depicts the outputs from the final VANISH code release.

*Figure 34: Final VANISH GUI and Main Camouflage Renderings*

Incorporated in the final VANISH code release is a section that displays a series of quantitative metrics of the camouflage effectiveness. This is accomplished through measurements of:

1. Pattern complexity,
2. Color fidelity,
3. Edge preservation,
4. Overall effectiveness score.

These metrics are visualized for the end user through radar charts and bar graphs, enabling operators/researchers with the ability to compare different camouflage modes and optimize them for specific environmental conditions or mission parameters. The scores are displayed on a 0 – 1 scale, with 1 being the best. Operators can assess their camouflage effectiveness over time on the Metrics Dashboards, using the data to drive more advanced tactics, techniques and procedures (TTPs) over time (see Figure 35).

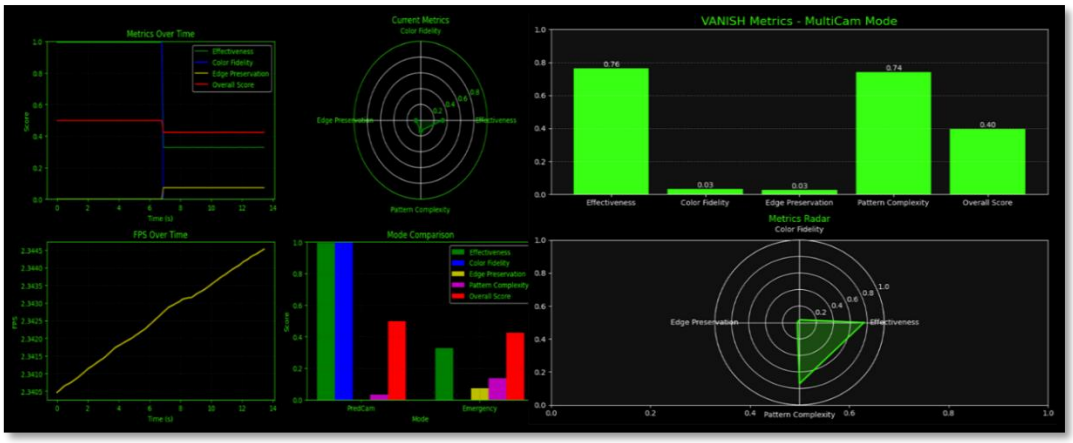



*Figure 35: VANISH Metrics Dashboard Output*

### 4.3 Lab Test of VANISH Code

Part of the VANISH test Methodology covered in chapter 3 discussed the evaluation of VANISH in both Lab and Field testing environments. These events were critical milestones in the VANISH Praxis as they showcased the current status of the VANISH code and material integrations, as well as uncovering gaps and issues within the system architecture. The initial VANISH lab testing was conducted at the HoloscapeLED test facility in Orlando, Florida. These first tests marked an important step in combining LED smart materials with a real-time machine learning software instantiation. This section explains the outcomes of the lab experiments, the objectives of the tests, and the outcomes of the trials. The hardware components of this test consisted of the M6 6.25mm surface mounted HoloscapeLED panels that served as the photonic smart skin test platform. These panels, supplied by HoloscapeLED engineers, displayed the VANISH baseline code and produced a real-time video output.

### 4.3.1 Testing Framework and Evaluation

The initial tests in the HoloscapeLED lab used AI/ML algorithms explored in section 3.2 and section 4.2 to validate how the system would operate in controlled environments. The baseline objectives for the tests were to validate:

1. Perform real-time VANISH AI/ML code for image processing and visualization.

2. Perform hardware and software integration between the VANISH code and the LED panels.

3. Record video output from the VANISH codebase to the LED.

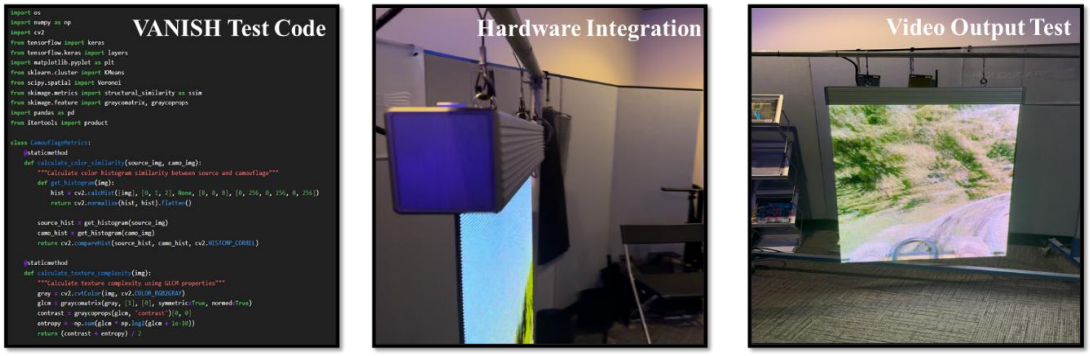



*Figure 36: Lab Test of the VANISH Framework*

### 4.3.2 Lab Test Results

The VANISH system laboratory tests at HoloscapeLED facility demonstrated satisfactory performance in integrating real-time VANISH code with the Holoscape M6 surface-mounted modules. The M6 modules' pixel density (25,600 pixels/m²) and brightness (≥5000 cd/m²) delivered a clear projection of camouflage patterns across variable lighting conditions (HoloscapeLED, 2024).

1. *VANISH AI/ML Code for Image Processing Results:* During this initial test, the VANISH code was still being developed, and in its current configuration, it

was able to produce a single video output. mp4 file that could be displayed on the HoloscapeLED screen. Processing time for the video took 43 seconds.

   a. *Note:* It took approximately 20 minutes to complete the hardware integrations between VANISH and HoloscapeLED panels due to a network configuration issue (see Section 4.3.3)

2. *Hardware and Software Integration Results*: After approximately 20 min of code processing and integration onto the HoloscapeLED display, the VANISH system was able to synchronize its video data outputs with the M6 modules, to display the complex video patterns from its outputs.
3. *Recording and Video Playback Results:* The M6 modules achieved their advertised 90% transparency, displaying the VANISH video output while demonstrating for the engineers the concept of platform adhesion.

*Table 13: VANISH Lab Test Results for Code*

| Parameter | Test Condition | Result | Implication |
|---|---|---|---|
| **Synthetic Pattern Projection** | Uniform lighting (500 lux) | Accurate pattern display across the entire video output | Effective in controlled indoor settings |
| **Pattern Transition Speed** | Video feed onto HoloscapeLED panel | Transition completed instantly once the video finished rendering in the codebase. The code took approximately 43 seconds to compile and display a video once integration was complete and tested. | Real-time adaptability for dynamic camouflage is still not achievable |
| **Energy Efficiency** | Continuous operation (1 hr.) | 370 watts/meter squared ($W/m^2$) average | Smart skin is suitable for extended tactical missions |
| **Brightness Consistency** | 50% ambient light | ≥5000 candelas per square meter ($cd/m^2$) achieved * | Effective in varying light conditions |

| | | | |
|---|---|---|---|
| **Operational Stability** | 85% relative humidity (RH), 75°F | No signal or model degradation was observed once the model loaded | Potentially reliable for humid and warm environments |
| **Transparency (Inactive State)** | Default configuration | 90% transparency maintained (M6 module) | Unobtrusive design when inactive |
| **Robustness of Integration** | High data feed (1 GB/s) achievable over Wi-Fi | No latency observed on the display once the code produced the output video and it was synced on the display. VANISH code took 43 seconds to render the video. Integration took approximately 20 min to achieve. | VANISH code will need additional work to render the video outputs faster for practical applications. |
| **Environmental Durability** | Simulated light and wind exposure | 100% operational | Withstands indoor operational stress factors. |

**Note: Candelas per square meter is a measure of brightness (also called luminance). It indicates how bright the display appears to the human eye. A brightness level of 5000 cd/m² signifies a very bright display suitable for environments with high ambient light* (HoloscapeLED, 2024).

### 4.3.3 Limitations and Challenges Identified

Despite the successful lab demonstration of VANISH's capabilities, the tests revealed several challenges and limitations:

1. *Dependency on Live Wi-Fi Connection:* A challenge of the current hardware system architecture is that it requires a steady Wi-Fi connection to stream the camouflage video data, as it cannot accept a direct HDMI input. In order to address this issue, the team was able to host live video streams locally on a laptop and cast them onto the screen via HoloscapeLEDs network. *Note: This workaround will not be practical for real-world combat operations.*

2. *Model Latency:* Processing times exhibited during the Lab Testing for VANISH were in excess of 43 seconds (see Figure 37). Before the field testing of VANISH, changes to the code baseline architecture will likely need to occur to minimize code latency and ensure the computer hardware can withstand the harsh environments of outdoor use.

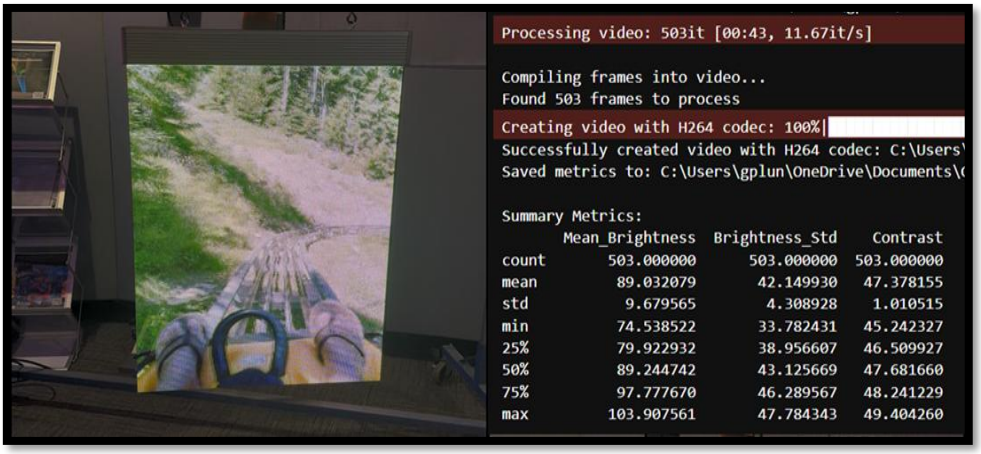



*Figure 37: VANISH Lab Test Video Output from VANISH*

### 4.3.4 Conclusion and Next Steps

Testing at HoloscapeLED labs provided the first validation that the VANISH codebase can run on flexible LED materials and display a complex pattern. The experiments conducted with HoloscapeLED engineers (see Figure 38) highlighted the scalability of the VANISH framework, as well as the adaptability of the photonic smart skins. Further testing

*Figure 38: HoloscapeLED Engineer (M2 Panel, 2024)*

will focus on expanding the environmental adaptability of the system, refining the baseline VANISH AI/ML algorithms/ approach for enhanced pattern complexity, and developing rapid integrations for real-world field deployments.

### 4.4 VANISH AI/ML Code Field Tests

On 12 January 2025, field testing of the VANISH codebase was performed to evaluate the real-world effectiveness of the VANISH AI/ML model under real-world conditions. The environments that served as the test locations were located within a 10-statute-mile radius of Apollo Beach, Florida, USA (latitude 27.773085 and longitude -82.4075931). Apollo Beach uniquely suits adaptive camouflage testing as the local environment provides multiple access points to urban landscapes, dense woodlands, beach/deserts, and marine settings.

*Table 14: Field Test Site Locations*

| # | Location | Latitude | Longitude | Environment / Description |
|---|---|---|---|---|
| **1** | **Beer Can Island** | 27.70 N | –82.38 W | Maritime and Desert Testing |
| **2** | **Publix Super Market** | 27.73 N | –82.43 W | Urban Landscapes |
| **3** | **Bull Frog Creek Park** | 27.80 N | –82.35 W | Fields and Forests |

Figure 39: shows a graphical depiction of these test sites:

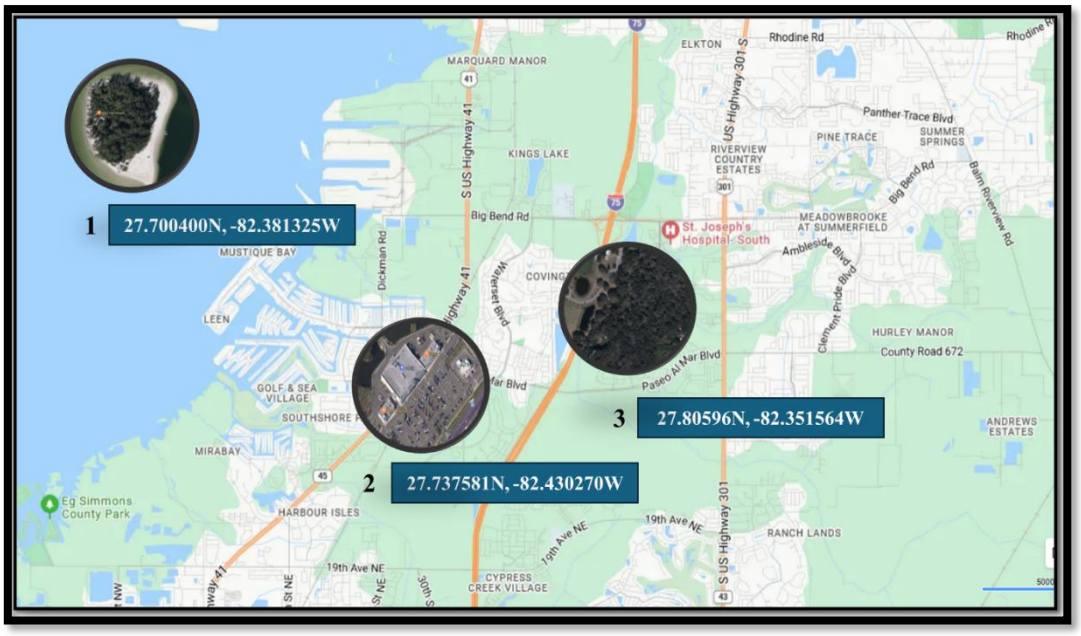



*Figure 39: VANISH Field Testing Locations (Microsoft, 2025)*

### 4.4.1 Field Test Employment

To test the VANISH codebase applications outside the HoloscapeLED laboratory setting, the physical architecture was transported to each of the various test sites by either a pickup truck (Bull Frog Creek Park and Publix Super Market) or via boat (Beer Can Island). The experiment used a DJI Mini 3 drone camera to capture the necessary VANISH video footage for ingestion into the model (see Figure 40). This was an expansion from the indoor testing, but it enabled the evaluation of how well the VANISH code could perform at creating active camouflage patterns when viewed from different heights and angles. The aerial perspective helped to assess how

*Figure 40: DJI Mini 3 for VANISH Video Input (DJI Corporation, 2025)*

effectively the camouflage patterns were able to blend with their surroundings under various lighting and background conditions.

Pre-planned flight paths were designed to capture direct sunlight, shadowed areas, and lateral perspectives from a predefined spot on the ground (see Figure 41). The resulting footage was then used to evaluate the system's optical camouflage performance once rendered through VANISH.

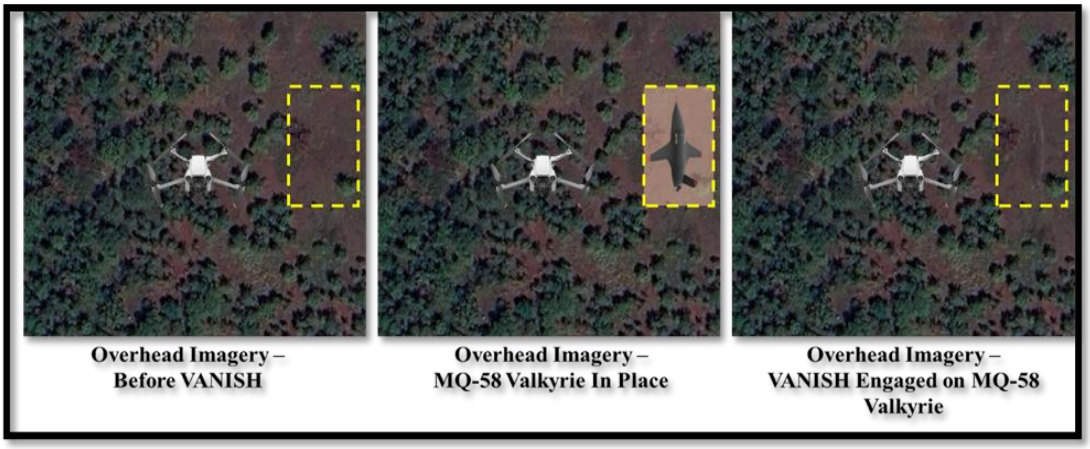



*Figure 41: DJI Mini Video Overwatch Architecture with MQ-58 Valkyrie Model*

#### 4.4.1.1 Beer Can Island (Maritime & Desert) Field Test Results

On the morning of 12 January 2025, the VANISH system was tested on Beer Can Island, near Apollo Beach, Florida. This test focused on combining maritime and desert video feed elements into the model from a single location. The environmental conditions presented dual challenges: intense sun glare from coastal waters and reflective (sandy) terrain mimicking desert-like conditions. The primary objective of the tests was to assess how quickly and accurately the VANISH code could adapt its camouflage patterns to significant shifts in natural lighting while maintaining operational stability under high temperatures and humidity.

*Table 15: VANISH Lab Test Results for Beer Can Island*

| Parameter | Lab Test Condition / Result | Beer Can Island (Maritime & Desert) |
|---|---|---|
| **Synthetic Pattern Projection** | **Uniform lighting (500 lux); Accurate pattern depiction.**<br><br>*Effective in controlled outdoor settings* | **Maritime:** Bright sunlight reflecting off water creates intense lighting conditions that went over 500 lux. Accurate pattern depiction. |
| | | **Desert:** Strong sunlight and reflective surfaces create challenging conditions that require the system to work with high degrees of accuracy. Accurate pattern depiction. |
| **Pattern Transition Speed** | **VANISH video rendering completed in 43 seconds** | **Maritime:** VANISH Code took 12.85 seconds to render. |
| | | **Desert:** VANISH Code took 10.25 seconds to render. |
| **Brightness Consistency** | **50% ambient light; ≥5000 cd/m²**<br><br>*Maintains clarity under various light levels* | **Maritime:** Model was able to overcome intense glare from water, preserving camouflage effectiveness. |
| | | **Desert:** Once the model rendered it was able to handle the demands of extremely bright conditions. |
| **Operational Stability** | **85% RH, 70F; No signal degradation**<br><br>*Model remained reliable in humid, warm climates* | **Maritime:** Salt and humidity are harsher than typical lab conditions; stable operation at 85% RH and 70F implies strong resilience. |
| | | **Desert:** High temperatures are common, so no degradation is crucial for continuous operation. |
| **Transparency (Inactive State Not Tested)** | **Default configuration; 70% transparency**<br><br>*Unobtrusive design* | **Maritime & Desert:** 70% transparency reduces the chance of detection when not active, especially beneficial in open, sunlit environments. |

#### 4.4.1.2 Beer Can Island Imagery Analysis

At Beer Can Island, the VANISH model underwent tests in a dual maritime and desert setting that challenged both its brightness and adaptive camouflage capabilities. The intense sunlight reflecting off water surfaces and sand required consistently high luminance (≥5000 cd/m²) to maintain optical accuracy. Images taken

during testing, and shown below, illustrate how well the system handled severe glare conditions. This field test was the first of the VANISH series, and the model's latency became quickly apparent as the system failed to adapt to changes in the environment in real-time, and was in a more near-real-time status of 10.25 seconds (desert) and 12.85 seconds (maritime) (see Figures 42 and 43).

```
Processing video: desert.mp4 (309 frames)
Optimizing parameters for desert_frame_0
Top 3 parameter combinations:
    n_colors  min_block_size  max_block_size  darkness_factor  \
74         5              20              25              1.0
56         5              10              25              1.0
65         5              15              25              1.0

    color_similarity  texture_complexity  edge_similarity  overall_score
74          0.463217           56.972778         0.007843      17.279473
56          0.512478           56.550860         0.007784      17.172584
65          0.396837           56.417713         0.009304      17.086840
Processed desert_frame_0 in 10.25 seconds
```

*Figure 42: Beer Can Island MultiCam (Desert) Code Processing Metrics*

```
Optimizing parameters for maritime_frame_48
Top 3 parameter combinations:
    n_colors  min_block_size  max_block_size  darkness_factor  \
56         5              10              25              1.0
68         5              15              30              1.0
29         4              10              25              1.0

    color_similarity  texture_complexity  edge_similarity  overall_score
56         -0.007808           18.992916         0.759429       5.922580
68         -0.007789           16.890005         0.800585       5.304061
29         -0.007091           16.562373         0.885345       5.231479
Processed maritime_frame_48 in 12.85 seconds
```

*Figure 43: Beer Can Island MultiCam (Maritime) Code Processing Metrics*

These visuals provide a frame of reference for the reliability of VANISH's synthetic pattern projection and potential for real-time adaptation that the system may be capable of delivering in demanding coastal and desert environments if the code

latency can be fixed (see Figures 44 and 45).

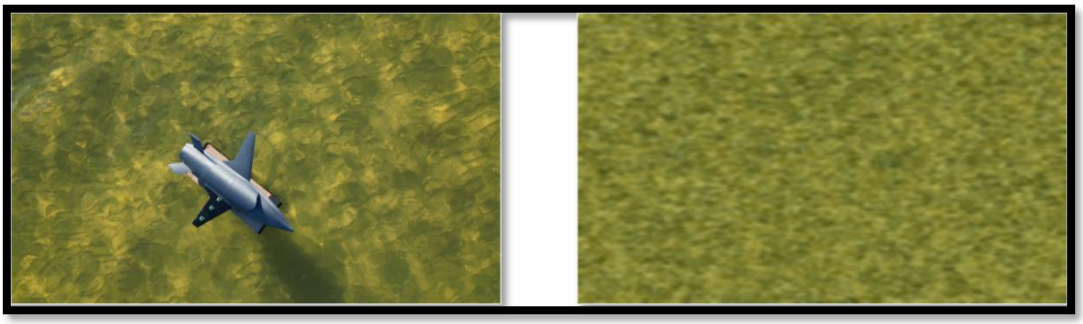


*Figure 44: Beer Can Island (Maritime) VANISH MultiCam Output*

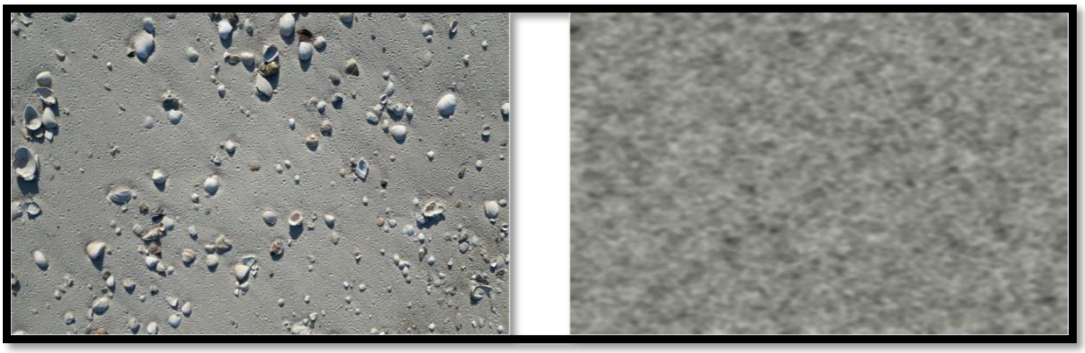


*Figure 45: Beer Can Island (Desert) VANISH MultiCam Output*

#### 4.4.1.3 Beer Can Island Test Summary

In both maritime and desert-like conditions, the system delivered consistent camouflage accuracy and robustness despite extreme brightness and changing scenery inputs. The code is still displaying an unsatisfactory level of latency with video and imagery outputs taking +10 seconds to render, but the system is demonstrating potential.

#### 4.4.2.1 Publix Supermarket (Urban) Field Test Results

On 12 January 2025, VANISH was deployed in an urban environment at a Publix Supermarket in Apollo Beach, Florida. During this test, the VANISH system faced a wide spectrum of ambient lighting conditions, from dimly lit loading docks to

exterior streetlights and vehicle headlights. City infrastructures often produce significant data noise and frequent, unpredictable shifts in illumination. The test aimed to evaluate how well the system maintained pattern accuracy in challenging high-glare conditions and what the associated latency might be with these evolutions. The ability to demonstrate rapid transitions in urban locations where discreet integration and concealment are paramount will be essential.

*Table 16: VANISH Lab Test Results for Publix Super Market*

| Parameter | Lab Test Condition / Result | Publix Supermarket (Urban) |
|---|---|---|
| **Synthetic Pattern Projection** | **Uniform lighting (500 lux); Accurate pattern depiction.**<br><br>*Effective in controlled outdoor settings* | **Outdoor Areas:** Typical store lighting is within the tested range but was not impacting the video output. |
| | | **Outdoor Facades:** Reflective glass and neon signs could be a problem due to high pattern accuracy. |
| **Pattern Transition Speed** | **VANISH video rendering completed in 43 seconds** | **Urban Environment:** Frequent, sudden changes in lighting (cars passing, store lights, streetlamps) were evident, but the model was running too slow to observe. |
| **Brightness Consistency** | **50% ambient light; ≥5000 cd/m²**<br><br>*Maintains clarity under various light levels* | **Urban Brightness Levels:** Model witnessed high luminance, but could handle glare from vehicle headlights; ensured patterns remain visible. |

| | | |
|---|---|---|
| **Environmental Durability** | **Simulated dust & high wind; 100% operational**<br><br>*Withstands moderate outdoor stresses* | **Urban Hazards:** Dust from construction sites, wind-driven debris, or high foot traffic needs to be tested further. |

#### 4.4.2.2 Publix Supermarket Imagery Analysis

Tested in an urban environment, the VANISH system faced frequent variations in artificial and ambient lighting, from neon signage and streetlights to passing vehicular traffic. This location showcased the system's ability to produce camouflage that was visually convincing in exposed urban settings. This field test was the second of the VANISH series, and the model's latency of 12.84 seconds was similar to that of Beer Can Island (see Figure 46).

```
Optimizing parameters for field_frame_92
Top 3 parameter combinations:
    n_colors  min_block_size  max_block_size  darkness_factor  \
56         5              10              25             1.00
28         4              10              25             0.95
2          3              10              25             1.00

    color_similarity  texture_complexity  edge_similarity  overall_score
56          0.700440           94.743150        -0.003509      28.702069
28          0.696513           93.082188         0.003645      28.204355
2           0.582300           90.697126         0.001306      27.442449
Processed field_frame_92 in 12.84 seconds
```

*Figure 46: Publix Supermarket MultiCam (Urban) Code Processing Metrics*

Photos from the test (see Figure 47) show the system's capacity for accurate transitions if the codebase can improve its latency.

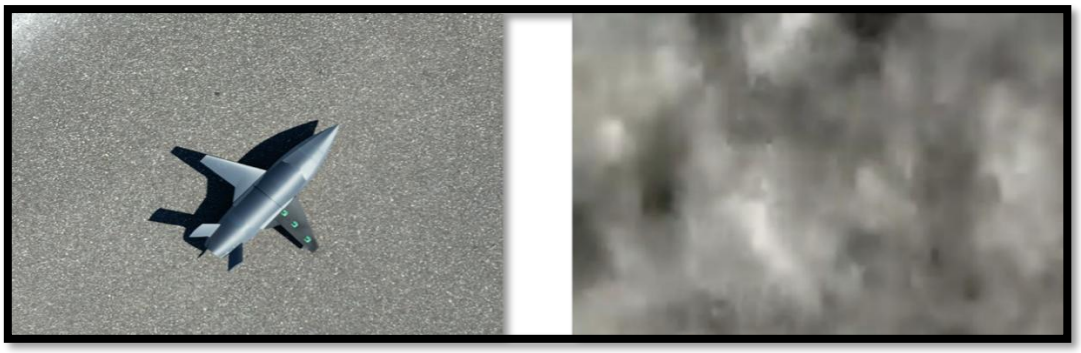


*Figure 47: Publix Super Market (Urban) VANISH MultiCam Output*

#### 4.4.2.3 Publix Supermarket Test Summary

In an active, light-variable city setting, the VANISH system maintained dependable performance, despite the model's latency concerns. Consistent high brightness was overcome by the models filtering and enabled the system to produce accurate renderings of its environment. Like Beer Can Island, the VANISH code is still displaying an unsatisfactory level of latency 12.84 seconds.

#### 4.4.3.1 Bull Frog Creek Park (Fields & Forests) Field Test Results

On 12 January 2025, at Bull Frog Creek Park, the VANISH codebase was evaluated in natural fields and forests, where patterns of light and shadow features are constantly in flux due to changing wind, foliage density, and terrain features. The test was designed to gauge the system's adaptability to accommodate sunlight, variable cloud cover, and other outdoor elements such as dust, pollen, and humidity that would be prevalent in forested locations.

*Table 17: VANISH Lab Test Results for Bull Frog Creek Park*

| Parameter | Lab Test Condition / Result | Bull Frog Creek Park (Fields & Forests) |
|---|---|---|

| | | |
|---|---|---|
| **Synthetic Pattern Projection** | **Uniform lighting (500 lux); high pattern accuracy**<br><br>*Effective in controlled indoor settings* | **Fields:** Variations in sunlight and shading were evident, once the model processed the video it was able to accommodate for these variables. |
| | | **Forests:** Variable lighting and leaf canopy shadows can be matched closely once the code rendered the video. |
| **Pattern Transition Speed** | **Environmental shift (~3 sec); Transition completed instantly once the video loaded**<br><br>*Latency issues persist, needs to transition faster.* | **Natural Terrain:** Rapid adaptation is crucial for shifting light conditions (e.g., clouds passing overhead or moving from open fields to wooded areas). Model rendered in 12.41 seconds. |
| **Brightness Consistency** | **50% ambient light; ≥5000 cd/m²**<br><br>*Maintains clarity under various light levels* | **Fields & Forests:** Canopy-filtered sunlight and variable open-sun conditions were able to be compensated for by VANISH, but further testing should be conducted. |
| **Operational Stability** | **85% RH, 75F; No signal degradation**<br><br>*Reliable in humid, warm climates* | **Natural Exposure:** Morning dew, rainfall, and warm daytime temperatures are common; no signal degradation ensures the system remains fully functional outdoors in ideal weather conditions, but further model testing is needed. |
| **Transparency (Inactive State)** | **Default configuration; 90% transparency**<br><br>*Unobtrusive design* | **Outdoor Stealth:** Inactive modules remain nearly invisible among natural backdrops, reducing the chance of detection by observers. |
| **Robustness of Integration** | **High data feed (1 GB/s); No latency observed once the video loaded**<br><br>*Seamless data flow after model renders* | **Complex Backgrounds:** High-bandwidth updates enable detailed pattern matching across varied terrains (grass, leaves, tree |

| | | |
|---|---|---|
| | | trunks) once the VANISH model loaded. |
| **Environmental Durability** | **Simulated dust & high wind; 100% operational**<br><br>*Withstands potential stresses in combat environments* | **Outdoor Conditions:** Wind, dust, pollen, and other natural elements may impede VANISH model operations. The system demonstrates resilience in open-air park environments, but austere weather conditions should be tested. |

#### 4.4.3.2 Bull Frog Creek Park Imagery Analysis

Tested at the local Bull Frog Park, VANISH contended well with fluctuating light levels caused by tree canopies, moving cloud cover, and open-field sunlight once the model rendered the output video. Contrary to the other environments tested, this location demanded a precise pattern blend to match the organic textures such as grass, bark, and foliage. This field test was the third and final of the VANISH series, and the model's latency of 12.41 seconds was similar to the previous two test sites (see Figure 48).

```
Processing video: field.MP4 (469 frames)
Optimizing parameters for field_frame_0
Top 3 parameter combinations:
    n_colors  min_block_size  max_block_size  darkness_factor  \
56         5              10              25              1.0
59         5              10              30              1.0
2          3              10              25              1.0

    color_similarity  texture_complexity  edge_similarity  overall_score
56          0.548086           87.062395         0.002209      26.338616
59          0.540279           82.756393         0.003221      25.043996
2           0.496053           80.507991         0.002678      24.351622
Processed field_frame_0 in 12.41 seconds
```

*Figure 48 Publix Bull Frog Creek Park MultiCam (Field) Code Processing Metrics*

The images captured during the demonstration reflect the system's smooth transitions between variable lighting spaces and its ability to sustain high pattern accuracy under scattered shadows (see Figure 49).

*Figure 49: Bull Frog Creek VANISH MultiCam Output*

#### 4.4.3.3 Bull Frog Creek Park Test Summary

During this test, the VANISH system maintained high levels of camouflage accuracy, but issues in latency and system response emerged when the tests switched between environments. In its current state the hardware and software appear to be practical enough for extended use in forested and open terrains, but the latency issues must be addressed. Like the previous locations, the VANISH code is still displaying an unsatisfactory level of latency with video outputs taking upwards of 12.41 seconds to process and display the images.

### 4.4.4 Field Test Limitations and Challenges

The VANISH codebase faces some significant challenges that require remediation. During the test when the surroundings would change rapidly across different domains, the codebase took too long to respond and produce its recommended camouflage output. This flaw in the baseline is a limiting factor for combat situations where reactions must happen instantly. The integration of hardware

and streamlining of computational resources presents a significant technical challenge that requires further refinement.

Future development efforts will need to focus on incorporating higher-resolution materials to enhance pattern quality and better leverage advanced AI/ML models to improve generalization across a larger range of operational conditions (e.g. rain). These upgrades will fix many current problems, but the baseline code requires significant modifications to solve the latency issues.

### 4.4.5 Conclusions and Next Steps

Field testing has confirmed that the VANISH codebase architecture is suitable for a wide range of environmental conditions if the codes latency can be fixed. The next steps will focus on addressing latency concerns in the code. Detailed playback analyses will continue in an effort to identify issues in the pattern transitions, brightness consistency, and overall “invisibility” in real world conditions. Overall, these combined efforts will hopefully guide future improvements.

# Chapter 5—Discussion and Conclusions

## 5.1 Discussion

This section of the Praxis focuses on answering research question 3 and the associated hypothesis, as it deals with the serious ethical concerns regarding the creation, use, and broader societal impacts of adaptive camouflage solutions like VANISH. As innovation progresses in the AI/ML field, the ethical implications reach well beyond the technology's direct applications. Researchers and government officials must consider how it could be used for both good and malicious purposes, and how to balance the fostering of new ideas while keeping the civilian populace safe. All this advancement sheds light on a bigger question of how future generations will see the world and interact with each other throughout everyday life. Security researcher Scharre pointed out in 2018 that “these aren't simple problems to solve”. Breakthroughs in the AI/ML spaces will continue to emerge, and ongoing dialogue across industry leaders regarding the ethical development and deployment of these technologies in sensitive applications, like adaptive camouflage, needs to be at the forefront of international debate (Scharre, 2018).

### 5.1.1 Dual-use Concerns and Potential Misuse

Dual-use technologies are those pieces of equipment and technology that can be used for both civilian and military applications. GPS is a common everyday tool that people use to navigate to and from stores, but also aids military platforms to navigate to and from strike locations. At present, governments carefully control the export of AI/ML technologies because they pose a strategic challenge for modern military forces. There are real concerns about what might happen if advanced camouflage systems fell into the wrong hands. Such technologies could potentially undermine public safety and destabilize international relations if not properly regulated. The potential for civilian misuse, such as privacy invasion, raises additional concerns about individual rights and societal well-being. These dangers show why a middle-ground approach, one that encourages new ideas while putting up strong safeguards against people who might use the technology to cause harm (Horowitz, AI and International Competition, 2019).

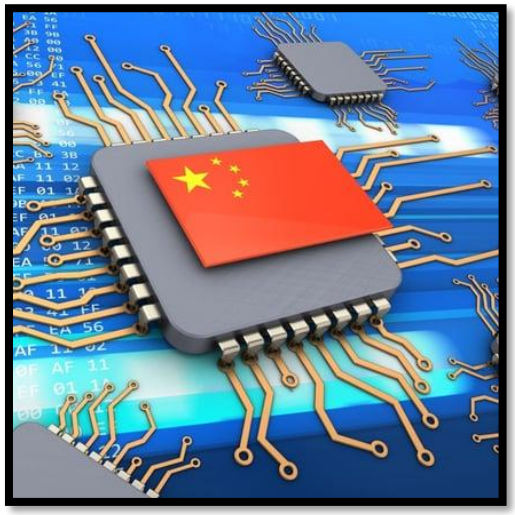


*Figure 50: AI Dual-Use Concerns Near Peer (ASPI Initiative, 2020)*

Companies working on AI-powered applications for military use, especially adaptive camouflage, should build in safety features from the outset. This might mean adding digital "fingerprints" that authorized agencies can detect, or intentional limitations that prevent the worst kinds of misuse. These safety features are similar to how some chemicals used in explosives contain tracers that help authorities, such as the FBI, identify where they came from if they're used illegally.

### 5.1.2 Privacy and Surveillance Implications

As smart camouflage systems like VANISH increase in their technical capabilities, questions about privacy versus safety in communities will naturally arise. When technology makes it easier for people or objects to blend into their surroundings, it could change what it means to be "anonymous" in public. When someone (or something) can be physically present but virtually invisible, it truly redefines reality. In the not too distant future, society will be entering a world where "hiding in plain sight" takes on a whole new meaning. Widespread adoption of advanced camouflage techniques could create an environment where constant vigilance becomes necessary. This could also accelerate more advanced counter-surveillance measures that normalize invasive monitoring systems, as currently witnessed in places like China, thereby creating a cycle that continuously encroaches upon civil liberties and personal privacy (The New York Times, 2022).

This raises important questions about public awareness when military technologies are deployed in urban areas. People and law enforcement should know when such systems are present and have some meaningful choice about their data being collected. As noted in research from 2022, consent becomes difficult when surveillance is invisible. Finding the right balance requires thoughtful updates to our legal protections and ethical standards, while developing technical solutions that respect individual privacy without compromising legitimate security needs (Alias, 2022). Judge Advocate Generals (JAGs) and other military policymakers must pay special attention to these technologies when drafting rules of engagement (ROE) and employment (ROEE) in future conflicts.

### 5.1.3 Legal and Regulatory Considerations

The development of effective governance for AI/ML-enhanced camouflage faces several hurdles, mainly brought on by the rapid pace of technological advancement, which often outstrips the laws and lawmakers who simply can't keep pace with how quickly technology evolves. Finding the right balance between encouraging new innovations and providing proper oversight to prevent harmful uses remains one of the key challenges in this field. The global nature of "open" technology development also complicates the creation of effective international regulations, while the inherent properties of camouflage technology make enforcement of any regulations particularly challenging (Scharre, 2018).

To address these challenges, several regulatory approaches could be considered:

1. Adaptive regulatory frameworks that can be quickly updated as technology evolves (Cath, 2017).
2. International cooperation in developing standards and agreements for the use of these technologies (Cath, 2017).
3. Incorporating ethics by design principles (as witnessed in the European Union General Data Protection Regulation (GDPR)) (Cath, 2017).
4. Implementing licensing and registration systems could help ensure responsible development and use (Cath, 2017).

Companies, such as OpenAI, that are involved in developing these leading technologies have implemented their own ethical policies and oversight mechanisms (OpenAI, 2024). This multi-faceted approach, which combines formal regulations with industry-led initiatives, could offer the best chance of creating a balanced and

effective governance framework that can adapt to the evolving landscape of VANISH or other AI/ML-enhanced camouflage technologies.

As Dr. Boulanin points out in IEEE Spectrum, the misuse of AI is “already happening” in civilian enterprises and “presents serious and insufficiently understood risks to international peace and security.” He notes that many “responsible AI” initiatives focus on civilian applications such as education and mobility but often fail to account for the potential weaponization of these toolsets. This omission is especially problematic given how easy it can be to adapt otherwise benign AI applications to harmful purposes (Ovink, 2022).

### 5.2 Conclusions

Across the VANISH Praxis, the research has sought to answer three primary questions:

1. **RQ 1:** Can AI/ML algorithms be used to develop adaptive camouflage models that can counter both human and computer vision detection?
2. **RQ 2:** What types of photonic smart-skin materials can best be paired with AI/ML models to create advanced camouflage for military applications?
3. **RQ 3:** What are potential adaptive camouflage technology applications and their associated ethical impacts across modern combat environments?

Over the course of this document, these questions have been examined in detail, and the prevailing theme across each of these topics’ points to ‘yes’, adaptive camouflage and its supporting models can be developed and employed across the modern battlefield, but to what extent they can defeat computer-aided models remains

uncertain. The long-term vision for adaptive camouflage in military applications is one of transformative potential that requires a dedicated focus on the ethical implications of this technology. As adaptive camouflage advances, the ability to remain undetected in increasingly complex environments will become critical cornerstone of military strategy.

Through the careful integration of advanced smart-skin materials, synthetic data, and autonomous systems such as VANISH, a new era of stealth capabilities is on the brink. However, as with all advanced military technologies, one must consider the ethical consequences of this technology. Dr. Oppenheimer and his team at Los Alamos radically changed the course of science and battlefield telemetry when they developed the atomic bomb; a similar course alteration may be on the brink with applied AI/ML applications.

### 5.3 Contributions to Body of Knowledge

The successful development of the VANISH AI/ML code baseline presented in Appendix A has for the first time, integrated into a single architecture many disparate concepts and models into a uniform and accessible ecosystem. For the military community, this academic endeavor represents a powerful first step in delivering applied AI for the everyday combatant. Soldiers, through Senior Leaders, are able to 'see' this technology in action and no longer perceive it as a black box that cannot be explained. VANISH looks to transform how the tactical, operational, and strategic levels of war are conducted. Forces that can minimize their optical signature and thereby reduce their risk of detection have a great deal of potential across the battlefield. Table 18 expounds on these implications and the potential outcomes of each:

*Table 18: Summary of VANISH Impacts on Military Operations*

| VANISH Advantages | Description | Implications | Potential Outcomes |
|---|---|---|---|
| **Tactical-Level Advantages** | Stealth | More informed decision-making; anticipation of enemy actions | Reduced casualties; more effective tactical maneuvers; and enhanced force protection |
| **Operational-Level Advantages** | Flexibility | Increased operational tempo; flexible deployment and redeployment | Greater responsiveness to evolving threats; decreased reliance on specialized equipment |
| **Strategic-Level Advantages** | Influential | Favors intelligence-driven, covert operations over large-scale engagements | More efficient use of resources; potential reduction in conflict scale and duration |
| **Impact on Enemy Forces** | Psychological | Potential deterrence: opponents forced to invest heavily in detection | Altered enemy decision-making; resource-intensive countermeasures |
| **Long-Term Evolution of Doctrine** | Transformational | NextGen stealth and minimal force requirements | A fundamental shift in military strategies, potentially reducing reliance on brute force |
| **Overall Strategic Implications** | Doctrine Shift | Enables smaller force employment to achieve large-scale effects, plus psychological impacts | Greater strategic flexibility; enhanced global power projection and influence |

Looking at the bigger Defense Tech picture, VANISH represents a complete rethinking of how traditional forces have approached camouflage. This technology

moves beyond the old-school fixed patterns designed for specific regions of the world or seasons. This shift toward adaptive systems that change in real-time holds the power to transform how the entire defense industry operates. This technology, driven largely by advancements in AI/ML computing power, is pushing defense-related industries to incorporate smarter materials, better algorithms, and connected ecosystems to accomplish missions more effectively.

### 5.4 Recommendations for Future Research

The future of VANISH and the concepts that will emerge from its implementation are vast. Imagine materials that not only change how they look, but also how they appear on thermal cameras or to radar systems. Future research will want to improve performance speeds, blend in other models to help better relate the camouflage to its surroundings and increase predictions during periods of high transition (e.g., sunrise, sunset). VANISH needs to improve its power consumption, and the photonic materials will need to be further ruggedized to withstand battlefield conditions.

Researchers should also look at ways to combine this technology with other military systems. Interoperability will be a huge component of further iterations of VANISH as more and more sensors are coupled with adaptive camouflage coatings. Another important area for future research is the integration of VANISH technology with other advanced military systems and platforms. This could include exploring how VANISH can be applied to different types of vehicles, drones, or even individual soldier equipment. Research should also examine how VANISH can be effectively combined with other emerging technologies such as augmented reality systems, advanced sensor networks, and autonomous decision-making algorithms. Such integration could potentially lead to the development of highly adaptive and

intelligent camouflage networks that can operate synergistically with other military assets to help them achieve their missions.

## References


Abadi, M. (2016, July 17). *TensorFlow.* Retrieved from usenix.org: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

Abdollahi, A. (2018, November 15). *Photochromic properties of stimuli-responsive cellulosic papers modified by spiropyran-acrylic copolymer in reusable pH-sensors.* Retrieved from https://www.sciencedirect.com: https://www.sciencedirect.com/science/article/abs/pii/S0144861718309408

Adobe Project Primrose. (2024, January 25). *Project Primrose: The story behind the digital dress-gone-viral.* Retrieved from https://research.adobe.com: https://research.adobe.com/news/project-primrose-the-story-behind-the-digital-dress-gone-viral/

Agarwal, K. (2023, June 01). *Quantum dots: an overview of synthesis, properties, and applications*. Retrieved from https://www.researchgate.net: https://www.researchgate.net/publication/371507100_Quantum_dots_an_overview_of_synthesis_properties_and_applications

Albright, T. (2023, October 17). *Adobe Primrose Unveils Next Gen Experience.* Retrieved from https://www.avnation.tv: https://www.avnation.tv/2023/10/17/adobe-primrose-unveils-next-gen-experience/

Alias. (2022, August 25). *Adaptive Urban Camouflage.* Retrieved from https://tradecrafting.substack.com/: https://tradecrafting.substack.com/p/adaptive-urban-camouflage

Army, U. (1999). *FM 20-3 CAMOUFLAGE, CONCEALMENT, AND DECOYS.* Washington DC: https://ia804605.us.archive.org/0/items/MManuals/Fm20-3CamouflageConcealmentAndDecoys.pdf.

ASPI Initiative. (2020, September 8). *Weaponizing the Belt and Road Initiative.* Retrieved from https://asiasociety.org: https://asiasociety.org/policy-institute/weaponizing-belt-and-road-initiative

Avellaneda, C. O. (2007, March 25). *Thermodynamic study of WO3 and WO3:Li+ thin films.* Retrieved from https://www.sciencedirect.com: https://www.sciencedirect.com/science/article/abs/pii/S0921510706002960

Ayoub, K. &. (2016, September 25). *Strategy in the Age of Artificial Intelligence.* Retrieved from Journal of Strategic Studies, 39(5-6), 793-819: https://doi.org/10.1080/01402390.2015.1088838

Bamwenda, G. R. (2001, March 9). *The visible light induced photocatalytic activity of tungsten trioxide powders.* Retrieved from www.sciencedirect.com: https://www.sciencedirect.com/science/article/abs/pii/S0926860X00007961

Bar-Shalom, Y. (2011, April 30). *Tracking Applications.* Retrieved from DOI.org: https://doi.org/10.1002/0471221279

Beaujuge, P. M. (2010). *Color control in π-conjugated organic polymers for use in electrochromic devices.* Retrieved from Chemical Reviews: https://doi.org/10.1021/cr900129a

Becker, M. (2019, December). *Privacy in the digital age: comparing and contrasting individual versus social approaches towards privacy.* Retrieved from https://www.researchgate.net: https://www.researchgate.net/publication/334522258_Privacy_in_the_digital_age_comparing_and_contrasting_individual_versus_social_approaches_towards_privacy

Bisoyi, H. K. (2016, June 01). *Light-Driven Materials.* Retrieved from DOI.org: https://doi.org/10.1021/acs.chemrev.6b00415

Bisoyi, H. K. (2021, December 23). *Liquid Crystals: Versatile Self-Organized Smart Soft Materials*. Retrieved from https://pubs.acs.org: https://pubs.acs.org/doi/epdf/10.1021/acs.chemrev.1c00761?ref=article_openPDF

Blackman, C. (2005, February 26). *Atmospheric Pressure Chemical Vapor Deposition of Crystalline Monoclinic WO3 and WO3-x Thin Films from Reaction of WCl6 with O-Containing Solvents and Their Photochromic and Electrochromic Properties.* Retrieved from https://www.semanticscholar.org/: https://www.semanticscholar.org/paper/Atmospheric-Pressure-Chemical-Vapor-Deposition-of-Blackman-Parkin/435084d9ca9cd02c151d7f01941212e5bce871ae

Bradski, G. (2000, May 01). *OpenCV Library.* Retrieved from opencv.org: https://opencv.org/about/

Brimley, W. &. (2014, July 27). *War in the Robotic Age.* Retrieved from CNAS.org: https://www.cnas.org/publications/reports/20yy-preparing-for-war-in-the-robotic-age

Buch, V. R. (2016). *Review on electrochromic property for WO3 thin films using different deposition techniques.* Retrieved from www.sciencedirect.com: https://www.sciencedirect.com/science/article/abs/pii/S2214785316300426

Businessner. (2023, November 23). *Adaptive Camouflage in Defense: The Science Behind Real-Time Color Changing Tech.* Retrieved from Businessner.com: https://businessner.com/adaptive-camouflage-in-defense-the-science-behind-real-time-color-changing-tech/

Businessner. (2023, November 10). *Adaptive Camouflage in Defense: The Science Behind Real-Time Color Changing Tech.* Retrieved from https://businessner.com: https://businessner.com/adaptive-camouflage-in-defense-the-science-behind-real-time-color-changing-tech/

Cath, C. (2017, March 28). *Artificial Intelligence and the 'Good Society': the US, EU, and UK approach.* Retrieved from https://pubmed.ncbi.nlm.nih.gov: https://pubmed.ncbi.nlm.nih.gov/28353045/

Chan, A. &. (2017, March 13). *AI and National Security.* Retrieved from Belfercenter.org: https:///publication/artificial-intelligence-and-national-security

Chang, L. &. (2017). Machine learning techniques for predicting enemy aircraft trajectories in air defense scenarios. . *Journal of Defense Analytics*, 9(4), 55-68.

Chollet, F. (2015, July 12). *Keras.* Retrieved from GITHUB.com: https://github.com/keras-team/keras

Colvin, V. L. (1994). *Light-emitting diodes made from cadmium selenide nanocrystals and a semiconducting polymer.* Retrieved from Nature: https://doi.org/10.1038/370354a0

Computer Hope. (2024, November 12). *Photoshop Gaussian Blur*. Retrieved from https://www.computerhope.com/: https://www.computerhope.com/jargon/p/photoshop-gaussian-blur.htm

Cummings. (2017, November 23). *AI and Future Warfare.* Retrieved from Chathamhouse.org: https://www.chathamhouse.org/sites/default/files/publications/research/2017-01-26-artificial-intelligence-future-warfare-cummings-final.pdf

Danine, A. (2014). *Room Temperature UV treated WO3 thin films for electrochromic devices on paper substrate.* Retrieved from www.sciencedirect.com: https://www.sciencedirect.com/science/article/abs/pii/S0013468614003302

Deb, S. K. (2008). *Opportunities and challenges in science and technology of WO3 for electrochromic and related applications.* Retrieved from Solar Energy Materials and Solar Cells: https://doi.org/10.1016/j.solmat.2007.08.011

DJI Corporation. (2025, January 07). *Mini-3 Specs*. Retrieved from https://www.dji.com: https://www.dji.com/mini-3/specs

Drzaic, P. S. (1995). *Liquid crystal dispersions.* Retrieved from Liquid Crystals: https://doi.org/10.1080/02678299508030599

Efros, A. L. (2016, August 03). *Origin and control of blinking in quantum dots*. Retrieved from https://pubmed.ncbi.nlm.nih.gov: https://pubmed.ncbi.nlm.nih.gov/27485584/

Elschner, A. (2010, November). *PEDOT: Principles and Applications of an Intrinsically Conductive Polymer.* Retrieved from https://www.researchgate.net: https://www.researchgate.net/publication/328611364_PEDOT_Principles_and_Applications_of_an_Intrinsically_Conductive_Polymer

Etsy. (18, October 2024). *Disruptive Pattern Material*. Retrieved from Etsy.com: https://www.etsy.com/accessories

Etsy. (2024, December 27). *MARPAT WOODLAND Digital Camo Military Spec*. Retrieved from Etsy.com: https://www.etsy.com/listing/1511765406/marpat-woodland-digital-camo-military?gpla=1&gao=1&&utm_source=bing&utm_medium=cpc&utm_campaign=shopping_us_ps--craft_supplies_and_tools-power_seller&utm_custom1=_k_6b1219bad2161010fa747e2bdf56bf53_k_&utm_content=bi

Etsy. (2025, January 08). *OCP Multicam Military Camouflage*. Retrieved from Etsy: https://www.etsy.com/listing/727734433/ocp-multicam-military-camouflage-nyco?click_key=fc9071518e5a24638555ecf0a44ce01c29c6e7be%3A727734433&click_sum=9eb78afb&ref=related-2&sts=1

Fonta, F. (2014, April 23). *Buffer size VS File size when reading binary file in Python.* Retrieved from https://stackoverflow.com: https://stackoverflow.com/questions/23234333/buffer-size-vs-file-size-when-reading-binary-file-in-python

Fractalcamo. (2025, January 22). *Swedish M90 Forest Camouflage Pattern*. Retrieved from www.fractalcamo.com: https://www.fractalcamo.com/store/p3086/Swedish_M90_Forest_Camouflage_Pattern.html

Galeev, D. (2020, June 18). *Learning High-Resolution Domain-Specific Representations with a GAN Generator.* Retrieved from www.deepai.org: https://deepai.org/publication/learning-high-resolution-domain-specific-representations-with-a-gan-generator

Gavim, A. E. (2017, December 01). *Influence of electrolyte distribution in PEDOT:PSS based flexible electrochromic devices.* Retrieved from https://www.sciencedirect.com: https://www.sciencedirect.com/science/article/abs/pii/S0009261417309478

Gearrack. (2025, January 25). *German Flecktarn Camouflage*. Retrieved from Ebay: https://www.ebay.com/sch/i.html?_nkw=german%20flecktarn%20camouflage&norover=1&mkevt=1&mkrid=711-34002-13078-0&mkcid=2&mkscid=102&keyword=german%20flecktarn%20camouflage&crlp=_&MT_ID=&geo_id=&rlsatarget=kwd-77309588614772:loc-190&adpos=&device=c&mktype=&l

Genevet, P. C. (2017). *Recent advances in planar optics: From plasmonic to dielectric metasurfaces.* Retrieved from DOI.org: https://doi.org/10.1364/OPTICA.4.000139

Girshick, R. (2015, April 18). *Fast R-CNN.* Retrieved from Openaccess.thecvf.com: https://openaccess.thecvf.com/content_iccv_2015/papers/Girshick_Fast_R-CNN_ICCV_2015_paper.pdf

Goodfellow, I. (2016, April 02). *MIT Press.* . Retrieved from Deep Learning: https://www.deeplearningbook.org/

Goodfellow, I. J. (2014, June 10). *Generative Adversarial Nets*. Retrieved from https://arxiv.org: https://arxiv.org/pdf/1406.2661

Granqvist, C. G. (2018). *Electrochromics on a roll: Web-coating and lamination for smart windows.* Retrieved from Surface and Coatings Technology: https://doi.org/10.1016/j.surfcoat.2017.07.037

Gregory R. Lee, R. G. (2019, April 12). *A Python package for wavelet analysis.* Retrieved from https://pywavelets.readthedocs.io: https://doi.org/10.21105/joss.01237

He, K. (2017, December 12). *Mask R-CNN.* Retrieved from ARXIV.org: https://arxiv.org/abs/1703.06870

HoloscapeLED. (2024, November 12). *M6 Surface Mounted Panel.* Retrieved from https://www.holoscapeled.com: https://www.holoscapeled.com/_files/ugd/25872f_fec4847d14b94920adc3053b8887cdf6.pdf

Horowitz, M. C. (2019, March 18). Retrieved from AI and International Competition: https://doi.org/10.15781/T2639KP49

Horowitz, M. C. (2019, March 01). *Artificial Intelligence, International Competition, and the Balance of Power.* Retrieved from Texas National Security Review, 3(1), 36-57. : https://doi.org/10.15781/T2639KP49

https://codetoflow.com/. (2025, March 29). *https://codetoflow.com/*. Retrieved from https://codetoflow.com/: https://codetoflow.com/

https://next-gen.materialsproject.org/. (2024, June 01). *https://next-gen.materialsproject.org/*. Retrieved from https://next-gen.materialsproject.org/: https://next-gen.materialsproject.org/

Huang, Y.-M. (2020, July 01). *Advances in Quantum-Dot-Based Displays.* Retrieved from https://www.researchgate.net: https://www.researchgate.net/publication/342739358_Advances_in_Quantum-Dot-Based_Displays

Hunan Haozhi Technology Co., Ltd. (2025, March 02). *Polymer Dispersed Liquid Crystal Smart PDLC Glass*. Retrieved from https://www.alibaba.com/product-detail/Polymer-Dispersed-Liquid-Crystal-Smart-PDLC_60743760809.html: https://www.alibaba.com/product-detail/Polymer-Dispersed-Liquid-Crystal-Smart-PDLC_60743760809.html

HyperStealth. (2025, January 9). *Quantum-Stealth*. Retrieved from https://www.hyperstealth.com: https://www.hyperstealth.com/Quantum-Stealth/index.html

Jang, E. (2010, July 21). *White-Light-Emitting Diodes with Quantum Dot Color Converters for Display Backlights*. Retrieved from https://onlinelibrary.wiley.com: https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.201000525?msockid=1ee3488f6adc69fb24075b526bd56860

Jianqin Yin, Y. H. (2011, January 01). *Detection of the Mobile Object with Camouflage Color Under Dynamic Background Based on Optical Flow* . Retrieved from https://www.sciencedirect.com/: https://pdf.sciencedirectassets.com/278653/1-s2.0-S1877705811X00115/1-s2.0-S1877705811019138/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjELD%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaCXVzLWVhc3QtMSJHMEUCIQCy%2BguvxMmZV4ym8UqUOZ6BAPzWk%2FqMEclBs3PqUbupsAIgasWUwHfd

John Schulman, F. W. (2017, August 28). *Proximal Policy Optimization Algorithms.* Retrieved from ARXIV.org: https://arxiv.org/abs/1707.06347

Jolliffe, I. T. (2002). *Springer.com.* Retrieved from Principal Component Analysis (PCA): https://link.springer.com/book/10.1007/b98835

Jongwon Kim, J. C. (01, March 2021). *RGDiNet: Efficient Onboard Object Detection with Faster R-CNN for Air-to-Ground Surveillance*. Retrieved from https://www.mdpi.com: https://www.mdpi.com/1424-8220/21/5/1677

Joseph, F. O. (2025, July 04). *Development of an Intelligent Cane for Visually Impaired Human Subjects*. Retrieved from www.researchgate.net: https://www.researchgate.net/publication/338585806_Development_of_an_Intelligent_Cane_for_Visually_Impaired_Human_Subjects

Kalman, R. E. (1960). *A new approach to linear filtering and prediction problems.* Retrieved from Journal of Basic Engineering: https://doi.org/10.1115/1.3662552

Khaleghi, B. (2013, May 03). *DOI.* Retrieved from Multisensor Data Fusion: https://doi.org/10.1016/j.inffus.2011.08.001

Khare. (2024, December 14). *Fourier Transform.* Retrieved from https://en.wikipedia.org: https://en.wikipedia.org/wiki/Fourier_transform

Kim, J. H. (2018). Deep learning-based target recognition for adaptive camouflage systems. *Journal of Military Science and Technology*, 40(3), 112-127.

Kortekaas, L. (2019, May 31). *The evolution of spiropyran: fundamentals and progress of an extraordinarily versatile photochrome.* Retrieved from https://pubs.rsc.org: https://pubs.rsc.org/en/content/articlehtml/2019/cs/c9cs00203k

Kott, A. (2019, April 12). *AI for Small Unit Maneuver.* Retrieved from DOI.org: https://doi.org/10.1109/MPOT.2019.2927909

Lagerwall, J. (2012). *A new era for liquid crystal research: Applications of liquid crystals in soft matter nano-, bio- and microtechnology.* Retrieved from Current Applied Physics: https://doi.org/10.1016/j.cap.2012.03.019

LeCun, Y. (2015). *Nature.* Retrieved from Deep Learning: https://doi.org/10.1038/nature14539

Lee. (2024, Ocotober 11). *CAMO (Camouflaged Object)*. Retrieved from https://paperswithcode.com: https://paperswithcode.com/dataset/camo

Lee, J. H. (2020, December 01). *QD Display: A Game-Changing Technology for the Display Industry*. Retrieved from www.researchgate.net: https://www.researchgate.net/figure/Schematic-picture-of-a-quantum-dot-QD-display_fig1_347001256

Lee-Thorp, J. (2021, May 09). *FNet: Mixing Tokens with Fourier Transforms.* Retrieved from https://arxiv.org: https://arxiv.org/abs/2105.03824

Li, H. G. (2020). *Liquid crystal devices for advanced photonic applications.* Retrieved from Advanced Optical Materials: https://doi.org/10.1002/adom.201902078

Lin, P. (2019, May 14). *Robot Ethics 2.0.* Retrieved from DOI.org: https://doi.org/10.1093/oso/9780190652951.001.0001

Liu. (2018, November 14). *Deep Learning in Target Recognition.* Retrieved from DOI.org: https://doi.org/10.1117/12.2304527

Lövenich, W. (2014, September). *PEDOT-properties and applications.* Retrieved from https://www.researchgate.net: https://www.researchgate.net/publication/264051560_PEDOT-properties_and_applications

Lumenalta. (2024, October 13). *Ethical considerations of AI.* Retrieved from https://lumenalta.com: https://lumenalta.com/insights/ethical-considerations-of-ai

Lyon, D. (2018, April 17). *Surveillance Capitalism, Surveillance Culture and Data Politics.* Retrieved from Data Politics, 64-77 : https://doi.org/10.4324/9781315167305-4

*M2 Panel.* (2024, December 07). Retrieved from www.holoscapeled.com: https://www.holoscapeled.com/surface-mounted-series?lightbox=dataItem-m1ayholg1

Mahulikar. (2016, July 11). *Infrared Signature Studies.* Retrieved from DOI.org: https://doi.org/10.1016/j.paerosci.2015.11.001

Mandal, M. (2022, October 11). *Spiropyran–Merocyanine Based Photochromic Fluorescent Probes: Design, Synthesis, and Applications*. Retrieved from https://pubs.acs.org/: https://pubs.acs.org/doi/10.1021/acsomega.2c04969

Meinzer, N. B. (2014). *Plasmonic meta-atoms and metasurfaces.* Retrieved from Nature Photonics: https://doi.org/10.1038/nphoton.2014.24

Mendes, C. (2019, April 20). *The $5 billion army camouflage that failed to hide its soldiers.* Retrieved from www.warhistoryonline.com: https://www.warhistoryonline.com/instant-articles/army-camouflage-that-cost-more.html

Merilaita, S. &. (2009, April 14). *Animal Camouflage.* Retrieved from DOI.org: https://doi.org/10.1098/rstb.2008.0217

Microsoft. (2025, January 03). *Apollo Beach*. Retrieved from www.Bing.com: https://www.bing.com/maps/?cp=27.795413%7E-82.350548&lvl=16.7&style=h

Minhao Wan, D. Z. (2022, November 18). *Combining Max pooling-Laplacian theory and k-means clustering for novel camouflage pattern design*. Retrieved from https://pmc.ncbi.nlm.nih.gov/: https://pmc.ncbi.nlm.nih.gov/articles/PMC9715738/

Minh-Quan Le, M. T. (2023, August 29). *CamoFA: A Learnable Fourier-based Augmentation for Camouflage Segmentation*. Retrieved from https://www.semanticscholar.org/: https://www.semanticscholar.org/paper/CamoFA%3A-A-Learnable-Fourier-based-Augmentation-for-Le-Tran/b71755f9112b28fe0945ace7af556850f4a81072

Mnih, V. (2015, November 03). *DOI.* Retrieved from Human-Level Control Through Deep RL: https://doi.org/10.1038/nature14236

Mnih, V. K. (2015, February 25). *Human-level control through deep reinforcement learning.* Retrieved from Nature.com: https://www.nature.com/articles/nature14236

Mortimer, R. J. (2011). *Electrochromic materials.* Retrieved from Annual Review of Materials Research: https://doi.org/10.1146/annurev-matsci-062910-100341

Ng, A. (2024, March 14). *Stanford Machine Learning Specialization.* Retrieved from www.coursera.org: https://www.coursera.org/specializations/machine-learning-introduction?utm_medium=sem&utm_source=bg&utm_campaign=b2c_emea_machine-learning-introduction_deeplearning-ai_ftcof_specializations_arte_jan_24_dr_geo-multi_sem_rsa_bing_lg-all&campaignid=662628110

O'Neill. (2020, August 11). *Active Camouflage Technology.* Retrieved from DOI.org: https://doi.org/10.1177/1548512919896855

OpenAI. (2024, September). *OpenAI Charter.* Retrieved from https://openai.com: https://openai.com/charter/

OpenAI. (2024, December 15). VANISH. *ChatGPT [Large language model]*. . https://chatgpt.com, Tampa .

Ortiz, S. (2023, June 28). *AI arms race: This global index ranks which nations dominate AI development.* Retrieved from https://www.zdnet.com: https://www.zdnet.com/article/ai-arms-race-this-global-index-ranks-which-nations-are-dominating-ai-development/

Ovink, V. B. (2022, August 27). *Civilian AI Is Already Being Misused by the Bad Guys.* Retrieved from https://spectrum.ieee.org/: https://spectrum.ieee.org/responsible-ai-threat

Papaefthimiou, S. (2001, April 2). *Advanced electrochromic devices based on WO3 thin films.* Retrieved from www.sciencedirect.com: https://www.sciencedirect.com/science/article/abs/pii/S0013468601003930

Pedram. (2013, August 20). *Blending Images while maintaining opacity.* Retrieved from https://stackoverflow.com: https://stackoverflow.com/questions/14582269/blending-images-while-maintaining-opacity

Pike. (2018, April 28). *Quantifying Camouflage.* . Retrieved from DOI.org: https://doi.org/10.1111/2041-210X.13019

Priya, S. (2024, July 16). *Tailoring lyotropic liquid crystals for skin barrier penetration: Exploring composition and structure–function relationships.* Retrieved from https://doi.org: https://doi.org/10.1063/5.0204909

Qing, Y. (2023, March 31). *Two-Way Generation of High-Resolution EO and SAR Images via Dual Distortion-Adaptive GANs*. Retrieved from https://www.mdpi.com: https://www.mdpi.com/2072-4292/15/7/1878

Rad, J. K. (2022, June 01). *Spiropyran-based advanced photoswitchable materials: A fascinating pathway to the future stimuli-responsive devices*. Retrieved from https://www.sciencedirect.com/: Spiropyran-based advanced photoswitchable materials: A fascinating pathway to the future stimuli-responsive devices

Ram, J. (2019, July 01). *Development of WO3-PEDOT: PSS hybrid nanocomposites based devices for liquefied petroleum gas (LPG) sensor*. Retrieved from https://www.researchgate.net/: https://www.researchgate.net/figure/Schematic-diagram-of-WO3-PEDOT-PSS-hybrid-composite-film-prepared-on-glass-substrate-by_fig5_334113909

Redmon, J. (2016, June 15). *ARXIV.org.* Retrieved from You Only Look Once (YOLO): https://arxiv.org/abs/1506.02640

Redmon, J. (2016, December 12). *You Only Look Once: Unified, Real-Time Object Detection.* Retrieved from ieeexplore.ieee.org: https://ieeexplore.ieee.org/document/7780460

Rizzuto, C. (2023, June 30). *Development of Electrochromic Devices, Based on Polymeric Gel, for Energy Saving Applications*. Retrieved from www.MDPI.com: https://www.mdpi.com/2073-4360/15/16/3347

Sameer. (2024, October 21). *Types of Plots: Visualization from Concept to Code*. Retrieved from https://www.analyticsvidhya.com: https://www.analyticsvidhya.com/blog/2021/12/12-data-plot-types-for-visualization/

*SANDF 2000 Soldier Camo* . (2014, August 16). Retrieved from MODDB.com: https://www.moddb.com/members/ironclad2014/images/sandf-2000-soldier-camo

Scharre, P. (2018, January 19). *Army of None: Autonomous Weapons and the Future of War.* Retrieved from Norton & Company: https://wwnorton.com/books/9780393608984

Scikit Image. (2024, November 10). *GLCM Texture Features*. Retrieved from https://scikit-image.org: https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_glcm.html

SemanticScholar. (2025, March 02). *Multispectral Image Classification Using PCA*. Retrieved from SemanticScholar.org: https://www.bing.com/images/search?view=detailV2&ccid=SQr8HhiA&id=95D1284AE3660BF4E6CB4AC31A6DF6A4ECF6B6C7&thid=OIP.SQr8HhiAcoCoWV75XSmXcAHaGv&mediaurl=https%3a%2f%2fai2-s2-public.s3.amazonaws.com%2ffigures%2f2017-08-08%2f5c24ffccd6efaf9ac480f7c3842eecf20

Seungmin Nam, D. W. (2023, April 27). *Biomimetic Multicolor-Separating Photonic Skin using Electrically Stretchable Chiral Photonic Elastomers.* Retrieved from onlinelibrary.wiley.com: https://onlinelibrary.wiley.com/doi/10.1002/adma.202302456

Sharkey. (2019, January 07). *AI Impact on Military Defence.* Retrieved from DOI.org: https://doi.org/10.1038/s42256-019-0095-3

Shi, Y. (2019, January 1). *Designed growth of WO3/PEDOT core/shell hybrid nanorod arrays with modulated electrochromic properties.* Retrieved from https://www.sciencedirect.com: https://www.sciencedirect.com/science/article/abs/pii/S1385894718316371

Shirasaki, Y. S. (2013). *Emergence of colloidal quantum-dot light-emitting technologies.* Retrieved from Nature Photonics: https://doi.org/10.1038/nphoton.2012.328

Smart Glass World. (2025, March 02). *What is Liquid Crystal on Silicon?* Retrieved from https://www.smartglassworld.ne: https://www.smartglassworld.net/liquid-crystal-on-silicon

Stackexchange.com. (2017, October 17). *OpenCV motion detection and tracking.* Retrieved from https://codereview.stackexchange.com: https://codereview.stackexchange.com/questions/178121/opencv-motion-detection-and-tracking

Statistics How To. (2025, July 04). *Shannon Entropy*. Retrieved from https://www.statisticshowto.com: https://www.statisticshowto.com/shannon-entropy/

Sukeru. (2025, January 22). *Australian Multicam Camouflage Uniform Ripstop Cotton*. Retrieved from sukeru.net: https://sukeru.net/?pid=177315579

Sutherland, R. L. (2003). *Electrically switchable holographic PDLC materials for displays and switchable lenses.* Retrieved from Journal of Applied Physics: https://doi.org/10.1063/1.1538368

Talas, L. (2019, September 01). *DOI.org.* Retrieved from Military Camouflage Evolution.: https://doi.org/10.1098/rstb.2019.0013

The New York Times. (2022, June 22). *How China's Surveillance Is Growing More Invasive* . Retrieved from www.Youtube.com/Visual Investigations: https://www.bing.com/videos/riverview/relatedvideo?q=chine+surviallnace&mid=94D48FB013F523CF1FCF94D48FB013F523CF1FCF&FORM=VIRE

The PyWavelets Developers. (2025, March 02). *PyWavelets - Wavelet Transforms in Python*. Retrieved from https://pywavelets: https://pywavelets.readthedocs.io/en/latest/

Tourani, A. (2020, May 28). *Motion detection by Mixture of Gaussian (MOG) background subtraction*. Retrieved from https://github.com/: https://github.com/GuilanITS/mog-motion-detection

Tutorials Point. (2025, January 12). *Scikit Image - GLCM Texture Features*. Retrieved from https://www.tutorialspoint.com/: https://www.tutorialspoint.com/scikit-image/scikit-image-glcm-texture-features.htm#:~:text=The%20skimage.feature.graycomatrix%20%28%29%20function%20is%20used%20for%20calculating,input%20image%2C%20which%20should%20be%20of%20integer%20type.

US Department of the Army. (1959). *Field Manual (FM) 101-10-1, Staff Officers' Field Manual Organization, Technical, and Logistical Data.* Arlington: Army Publishing Directorate.

Verbruggen, B. &. (2017, December 16). *Autonomy in Weapon Systems.* Retrieved from SIPRI.org: https://www.sipri.org/publications/2017/other-publications/mapping-development-autonomy-weapon-systems

Vincent Boulanin, M. V. (2017, December). *ARTICLE 36 REVIEWS Dealing with the challenges posed by emerging technologies.* Retrieved from https://www.sipri.org: https://www.sipri.org/sites/default/files/2017-12/article_36_report_1712.pdf

vitalflux.com. (2021, 11 01). *CNN Architecture.* Retrieved from https://vitalflux.com: https://vitalflux.com/wp-content/uploads/2021/11/VGG16-CNN-Architecture.png

Vollmer, M. &. (2017, February 25). *Infrared Imaging.* Retrieved from DOI.org: https://doi.org/10.1002/9783527693306

Wang, L. (2019). *Learning Parallax Attention for Stereo Image Super-Resolution.* Retrieved from https://openaccess.thecvf.com/: https://openaccess.thecvf.com/content_CVPR_2019/papers/Wang_Learning_Parallax_Attention_for_Stereo_Image_Super-Resolution_CVPR_2019_paper.pdf

Wanying Zhang, Z. S. (2022, Ocotober 31). *Recent progress on PEDOT-based wearable bioelectronics.* Retrieved from https://onlinelibrary.wiley.com: https://onlinelibrary.wiley.com/doi/10.1002/VIW.20220030?msockid=1ee3488f6adc69fb24075b526bd56860

White House. (2024, August 14). *2024-Agency-AI-Reporting-per-EO-14110.* Retrieved from www.whitehouse.gov: https://www.whitehouse.gov/wp-content/uploads/2024/08/Instructions-for-2024-Agency-AI-Reporting-per-EO-14110.pdf

Wikipedia. (2024, December 16). *Predator*. Retrieved from www.wikipedia.com: https://en.wikipedia.org/wiki/Predator_(film)

Wikipedia. (2025, June 27). *https://en.wikipedia.org*. Retrieved from Gaussian Blur: https://en.wikipedia.org/wiki/Gaussian_blur

Wikipedia. (2025, March 13). *K-means Clustering*. Retrieved from https://en.wikipedia.org: https://en.wikipedia.org/wiki/K-means_clustering

*Wikipedia.org.* (2024, November 23). Retrieved from Q-Learning: https://en.wikipedia.org/wiki/Q-learning

Xiao, H. (2020, July 30). *Fast Self-Adaptive Digital Camouflage Design Method Based on Deep Learning*. Retrieved from https://www.mdpi.com/: https://www.mdpi.com/2076-3417/10/15/5284

Y. Yifat, M. E. (2015, March 02). *Highly Efficient and Broadband Wide-Angle Holography Using Patch-Dipole Nanoantenna Reflectarrays*. Retrieved from Nano Photonic: https://nanophotonics.sites.tau.ac.il/?page_id=124

Yu, C. (2014, August 18). *Adaptive optoelectronic camouflage systems with designs inspired by cephalopod skins.* Retrieved from https://www.pnas.org: https://www.pnas.org/doi/10.1073/pnas.1410494111

Yu, P. (2023, February 01). *Polymer-stabilized cholesteric liquid crystal films with broadband reflection formed by photomask polymerization.* Retrieved from https://doi.org: https://doi.org/10.1016/j.optmat.2022.113385

Zhang, W. (2025, July 04). *K-Means*. Retrieved from https://wei2624.github.io: https://wei2624.github.io/MachineLearning/usv_kmeans/

Zuboff, S. (2019, August 23). *The Age of Surveillance Capitalism.* Retrieved from Profile Books: https://profilebooks.com/work/the-age-of-surveillance-capitalism/

# Appendix A

## A.1 Introduction

This appendix contains the baseline instantiation of the VANISH codebase. This final code release represents a series of iterative code development efforts that spanned from August 2024 to May 2025. The elements within the code built upon the research and lessons learned across the entirety of this Praxis and display two fundamental camouflage techniques:

1. *The VANISH PredCam Model*
2. *The VANISH MultiCam Model*

Further development and series releases of the VANISH code baseline will undoubtedly occur as future research in this evolving topic emerge.

### A.1.1 VANISH Codebase

VANISH serves as the core engine of the overall VANISH application. This code builds upon the baseline models to deliver a robust capability ready for real-world deployment.

```
import cv2
import numpy as np
import ipywidgets as widgets
from IPython.display import display, HTML, clear_output
import matplotlib.pyplot as plt
import base64
import time
import threading
import io
from PIL import Image as PILImage
from PIL import ImageFont, ImageDraw

# Define PREDATOR  GREEN color for consistency
PREDATOR_GREEN = (20, 255, 60)

# Create output areas
image_widget = widgets.Image()
```

```
debug_output = widgets.Output()
metrics_output = widgets.Output()

# Create UI elements
start_button = widgets.Button(description="Start Camera", button_style='success')
stop_button = widgets.Button(description="Stop Camera", button_style='danger')
show_metrics_button = widgets.Button(description="Show Metrics", button_style='info')
force_change_button = widgets.Button(description="Force Mode Change", button_style='warning')
mode_dropdown = widgets.Dropdown(
    options=['No Camouflage', 'MultiCam', 'PredCam', 'Emergency'],
    value='No Camouflage',
    description='Mode:',
    disabled=False,
)
status_label = widgets.Label(value="Camera not started")

# Apply custom CSS for Raleway font and predator green color
custom_css = """
<style>
@import url('https://fonts.googleapis.com/css2?family=Raleway:wght@400;700&display=swap');

.widget-label, .widget-dropdown, .jp-OutputArea-output, h2 {
    font-family: 'Raleway', sans-serif !important;
    color: #39FF14 !important; /* Predator lime green */
}

/* Make button text black for better readability */
.widget-button {
    font-family: 'Raleway', sans-serif !important;
    color: #000000 !important; /* Black text for buttons */
    font-weight: bold !important;
}

.jp-OutputArea-output pre {
    font-family: monospace !important;
    color: #39FF14 !important;
}

/* Custom title styling */
.vanish-title {
    font-family: 'Raleway', sans-serif !important;
    color: #39FF14 !important;
    font-weight: 700 !important;
    text-shadow: 0 0 5px #39FF14, 0 0 10px #39FF14 !important;
    letter-spacing: 2px !important;
}

/* Make the entire background black */
body {
    background-color: #000000 !important;
}
</style>
"""

# Global variables
camera_running = False
image_data = None
metrics_data = {}  # For storing calculated metrics
current_processing_mode = None  # Track the mode that's being processed
```

```
def calculate_metrics(original_frame, processed_frame, mode):
    """Calculate various metrics between original and processed frames"""
    # Effectiveness - difference between frames
    diff = cv2.absdiff(original_frame, processed_frame)
    effectiveness = 1.0 - (np.sum(diff) / (diff.size * 255))

    # Color fidelity - histogram comparison
    if len(original_frame.shape) == 3:  # Color image
        hist1 = cv2.calcHist([original_frame], [0, 1, 2], None, [8, 8, 8], [0, 256, 0, 256, 0, 256])
        hist2 = cv2.calcHist([processed_frame], [0, 1, 2], None, [8, 8, 8], [0, 256, 0, 256, 0, 256])

        # Normalize histograms
        cv2.normalize(hist1, hist1, 0, 1, cv2.NORM_MINMAX)
        cv2.normalize(hist2, hist2, 0, 1, cv2.NORM_MINMAX)

        # Compare histograms
        color_fidelity = cv2.compareHist(hist1, hist2, cv2.HISTCMP_CORREL)
    else:
        color_fidelity = 0.5

    # Edge preservation - edge detection comparison
    gray1 = cv2.cvtColor(original_frame, cv2.COLOR_BGR2GRAY) if len(original_frame.shape) ==
3 else original_frame
    gray2 = cv2.cvtColor(processed_frame, cv2.COLOR_BGR2GRAY) if
len(processed_frame.shape) == 3 else processed_frame

    edges1 = cv2.Canny(gray1, 100, 200)
    edges2 = cv2.Canny(gray2, 100, 200)

    if np.sum(edges1) > 0:
        edge_preservation = np.sum((edges2 > 0) & (edges1 > 0)) / np.sum(edges1 > 0)
    else:
        edge_preservation = 1.0

    # Pattern complexity - entropy
    entropy = calculate_entropy(gray2)
    pattern_complexity = min(entropy / 7.0, 1.0)  # Normalize 0-1

    # Calculate overall score based on mode
    if mode == 'MultiCam':
        score = 0.5 * effectiveness + 0.3 * color_fidelity + 0.2 * edge_preservation
    elif mode == 'PredCam':
        score = 0.3 * effectiveness + 0.2 * color_fidelity + 0.5 * edge_preservation
    elif mode == 'Emergency':
        score = 0.6 * (1 - effectiveness) + 0.1 * color_fidelity + 0.3 * edge_preservation
    else:  # No Camouflage
        score = 0.5

    # Return metrics dictionary
    return {
        'mode': mode,
        'effectiveness': float(effectiveness),
        'color_fidelity': float(color_fidelity),
        'edge_preservation': float(edge_preservation),
        'pattern_complexity': float(pattern_complexity),
        'overall_score': float(score)
    }

def calculate_entropy(image):
    """Calculate Shannon entropy of an image"""
```

```
    histogram = cv2.calcHist([image], [0], None, [256], [0, 256])
    histogram = histogram.ravel() / histogram.sum()
    non_zero = histogram > 0
    return -np.sum(histogram[non_zero] * np.log2(histogram[non_zero]))

def display_metrics_charts():
    """Display metrics charts in the metrics output area"""
    global metrics_data

    with metrics_output:
        clear_output(wait=True)

        if not metrics_data:
            print("No metrics data available yet.")
            return

        # Creates figure with predator theme mask
        plt.style.use('dark_background')
        fig, axs = plt.subplots(2, 1, figsize=(10, 8))
        fig.patch.set_facecolor('#000000')

        # Bar chart of current metrics
        metrics = ['Effectiveness', 'Color Fidelity', 'Edge Preservation', 'Pattern Complexity', 'Overall
Score']
        values = [
            metrics_data['effectiveness'],
            metrics_data['color_fidelity'],
            metrics_data['edge_preservation'],
            metrics_data['pattern_complexity'],
            metrics_data['overall_score']
        ]

        # Plot 1: Bar chart
        bars = axs[0].bar(metrics, values, color='#39FF14')

        # Add value labels
        for bar in bars:
            height = bar.get_height()
            axs[0].text(bar.get_x() + bar.get_width()/2., height,
                    f'{height:.2f}',
                    ha='center', va='bottom', color='white')

        axs[0].set_title(f'VANISH Metrics - {metrics_data["mode"]} Mode', color='#39FF14',
fontsize=16)
        axs[0].set_ylim(0, 1)
        axs[0].grid(axis='y', linestyle='--', alpha=0.3)
        axs[0].set_facecolor('#111111')

        # Plot 2: Radar chart for metrics
        # Create radar chart using polar plot
        ax = axs[1]
        ax.set_facecolor('#111111')

        # Create radar chart categories and values
        categories = ['Effectiveness', 'Color Fidelity', 'Edge Preservation', 'Pattern Complexity']
        radar_values = [
            metrics_data['effectiveness'],
            metrics_data['color_fidelity'],
            metrics_data['edge_preservation'],
            metrics_data['pattern_complexity']
```

```
        ]

        # Polygon
        categories = categories + [categories[0]]
        radar_values = radar_values + [radar_values[0]]

        # Compute angle for each category
        angles = np.linspace(0, 2*np.pi, len(categories), endpoint=True)

        # Plot data
        ax = plt.subplot(2, 1, 2, polar=True)
        ax.plot(angles, radar_values, linewidth=2, linestyle='solid', color='#39FF14')
        ax.fill(angles, radar_values, color='#39FF14', alpha=0.25)

        # Fill in axis labels
        ax.set_thetagrids(angles[:-1] * 180/np.pi, categories[:-1])
        ax.set_ylim(0, 1)
        ax.grid(True)
        ax.set_facecolor('#111111')

        # Title
        ax.set_title('Metrics Radar', color='#39FF14')

        plt.tight_layout()
        plt.show()

# Update image widget
def update_image_widget(frame):
    """Convert a frame to JPEG and update the image widget"""
    try:
        # Convert BGR to RGB
        if frame is not None and len(frame.shape) == 3:
            frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

            # Convert using PIL first (can be more stable than cv2.imencode)
            pil_img = PILImage.fromarray(frame_rgb)

            # Use BytesIO for in-memory file operations
            img_bytes = io.BytesIO()
            pil_img.save(img_bytes, format='JPEG', quality=90)
            img_bytes.seek(0)

            # Update the image widget
            image_widget.value = img_bytes.read()
            return True
    except Exception as e:
        with debug_output:
            print(f"Error updating image: {str(e)}")
        return False

# Function to create noise texture for MultiCam
def create_noise_texture(shape, scale=100):
    """Create simplified noise texture"""
    # Create random noise
    noise = np.random.randn(shape[0], shape[1])

    # Smooth it
    noise = cv2.GaussianBlur(noise, (0, 0), scale/10)

    # Normalize to [0, 1]
```

```
    noise = (noise - noise.min()) / (noise.max() - noise.min())
    return noise

# Function to extract dominant colors
def extract_dominant_colors(frame, n_colors=5, darkness_factor=0.95):
    """Extract dominant colors using simple color quantization"""
    # Downsample image for faster processing
    h, w = frame.shape[:2]
    small = cv2.resize(frame, (w//4, h//4))

    # Reshape and convert to float for processing
    pixels = small.reshape(-1, 3).astype(np.float32)

    # Use k-means clustering to find dominant colors
    criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0)
    _, labels, centers = cv2.kmeans(pixels, n_colors, None, criteria, 10,
cv2.KMEANS_RANDOM_CENTERS)

    # Convert back to uint8
    centers = centers.astype(np.uint8)

    # Sort by brightness
    brightness = np.mean(centers, axis=1)
    sorted_indices = np.argsort(brightness)

    # Apply darkness factor
    colors = []
    for idx in sorted_indices:
        color = centers[idx] * darkness_factor
        colors.append(tuple(color.astype(np.uint8)))

    return colors

# Apply effect based on mode - modified to add better error handling
def apply_effect(frame, mode):
    try:
        if mode == 'No Camouflage':
            return frame.copy()

        elif mode == 'PredCam':
            # Edge preserving filter for a camouflage-like effect
            with debug_output:
                print(f"Applying PredCam effect to frame shape: {frame.shape}")
            result = cv2.edgePreservingFilter(frame, flags=cv2.RECURS_FILTER, sigma_s=60,
sigma_r=0.4)
            return result

        elif mode == 'MultiCam':
            height, width = frame.shape[:2]

            # Default parameters
            pattern_complexity = 0.7
            noise_level = 0.05
            darkness_factor = 0.95

            with debug_output:
                print(f"Applying MultiCam effect to frame shape: {frame.shape}")

            # Extract dominant colors
            colors = extract_dominant_colors(frame, n_colors=5, darkness_factor=darkness_factor)
```

```
        # Generate noise layers at different scales
        noise1 = create_noise_texture((height//4, width//4), scale=40)
        noise2 = create_noise_texture((height//4, width//4), scale=20)
        noise3 = create_noise_texture((height//4, width//4), scale=10)

        # Resize to frame dimensions
        noise1 = cv2.resize(noise1, (width, height))
        noise2 = cv2.resize(noise2, (width, height))
        noise3 = cv2.resize(noise3, (width, height))

        # Combine noise layers with pattern complexity weighting
        combined = noise1 * 0.5 * pattern_complexity + noise2 * 0.3 * pattern_complexity + noise3
* 0.2 * pattern_complexity
        combined = (combined - combined.min()) / (combined.max() - combined.min())

        # Create pattern using thresholds
        thresholds = np.linspace(0, 1, len(colors) + 1)
        pattern = np.zeros((height, width, 3), dtype=np.uint8)

        # Fill pattern with colors based on noise thresholds
        for i in range(len(colors)):
            mask = (combined >= thresholds[i]) & (combined < thresholds[i + 1])
            mask = np.repeat(mask[:, :, np.newaxis], 3, axis=2)  # Expand mask to 3 channels
            pattern = np.where(mask, colors[i], pattern)

        # Add texture noise if needed
        if noise_level > 0:
            texture_noise = np.random.normal(0, noise_level * 255, (height, width,
3)).astype(np.int16)
            pattern = np.clip(pattern.astype(np.int16) + texture_noise, 0, 255).astype(np.uint8)

        # Apply a slight blur for more natural transitions
        pattern = cv2.GaussianBlur(pattern, (3, 3), 0)

        return pattern

    elif mode == 'Emergency':
        # Create high visibility orange pattern
        height, width = frame.shape[:2]
        with debug_output:
            print(f"Applying Emergency effect to frame shape: {frame.shape}")

        emergency = np.zeros_like(frame)
        emergency[:] = (0, 97, 255)  # Blaze orange in BGR

        # Add stripes
        for i in range(0, width + height, 80):
            thickness = 20
            cv2.line(emergency, (0, i), (i, 0), (255, 255, 255), thickness)

        # Add text
        cv2.putText(emergency, "EMERGENCY", (width//2-120, height//2),
                    cv2.FONT_HERSHEY_SIMPLEX, 1.5, (0, 0, 0), 4)

        return emergency

    # Default case - return unchanged
    return frame.copy()
```

```
    except Exception as e:
        # Log the error and return the original frame
        with debug_output:
            print(f"Error applying effect for mode {mode}: {str(e)}")
            import traceback
            traceback.print_exc()
        return frame.copy()

# Function to add text with predator-style font using OpenCV
def add_text_with_style(img, text, position, font_scale=0.7, thickness=2, use_glow=False):
    if use_glow:
        # Add glow effect first (larger, blurred text)
        glow_size = thickness + 2
        cv2.putText(img, text, position, cv2.FONT_HERSHEY_SIMPLEX, font_scale,
PREDATOR_GREEN, glow_size*2)
    # Add the main text on top
    cv2.putText(img, text, position, cv2.FONT_HERSHEY_SIMPLEX, font_scale,
PREDATOR_GREEN, thickness)

# Main camera processing function
def process_camera():
    global camera_running, metrics_data, current_processing_mode

    # Try to open camera
    with debug_output:
        print("Opening camera...")

    cap = cv2.VideoCapture(0)

    if not cap.isOpened():
        with debug_output:
            print("Default camera failed, trying alternatives...")

        for idx in range(1, 4):
            cap = cv2.VideoCapture(idx)
            if cap.isOpened():
                with debug_output:
                    print(f"Camera at index {idx} opened successfully")
                break

    if not cap.isOpened():
        with debug_output:
            print("Failed to open any camera")
        status_label.value = "No camera found"
        start_button.disabled = False
        stop_button.disabled = True
        mode_dropdown.disabled = False
        current_processing_mode = None
        return

    # Set lower resolution for better performance
    cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
    cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

    # Allow camera to initialize
    time.sleep(0.5)

    # Read test frame
    ret, frame = cap.read()
    if not ret:
```

```
            with debug_output:
                print("Failed to read frame")
            cap.release()
            start_button.disabled = False
            stop_button.disabled = True
            mode_dropdown.disabled = False
            current_processing_mode = None
            return

        with debug_output:
            print(f"Camera initialized, frame size: {frame.shape}")

        # FPS tracking
        frame_count = 0
        start_time = time.time()
        display_count = 0
        display_time = time.time()

        try:
            # Main processing loop
            while camera_running:
                # Read frame
                ret, frame = cap.read()
                if not ret:
                    with debug_output:
                        print("Failed to read frame, retrying...")
                    time.sleep(0.1)
                    continue

                # Get current mode from dropdown - check if mode has changed as this has been an issue
                target_mode = mode_dropdown.value
                if current_processing_mode != target_mode:
                    with debug_output:
                        print(f"Mode changed: {current_processing_mode} -> {target_mode}")
                    current_processing_mode = target_mode

                # Process based on selected mode
                try:
                    with debug_output:
                        print(f"Processing frame with mode: {current_processing_mode}")
                    processed = apply_effect(frame, current_processing_mode)
                    # Calculate metrics
                    metrics_data = calculate_metrics(frame, processed, current_processing_mode)
                except Exception as e:
                    with debug_output:
                        print(f"Error processing frame with mode {current_processing_mode}: {str(e)}")
                        import traceback
                        traceback.print_exc()
                    # Fallback to original frame if effect fails
                    processed = frame.copy()
                    metrics_data = {'mode': current_processing_mode, 'effectiveness': 0.0,
                                    'color_fidelity': 0.0, 'edge_preservation': 0.0,
                                    'pattern_complexity': 0.0, 'overall_score': 0.0}

                # Create side-by-side view with labels
                h, w = frame.shape[:2]
                combined = np.hstack((frame, processed))

                # Add separator line
                cv2.line(combined, (w, 0), (w, h), PREDATOR_GREEN, 2)
```

```
            # Add styled text labels without glow effect
            add_text_with_style(combined, "Original", (10, 30), use_glow=False)
            add_text_with_style(combined, current_processing_mode, (w + 10, 30), use_glow=False)

            # Add system status in predator style
            add_text_with_style(combined, "VANISH SYSTEM ACTIVE", (w-180, h-40), 0.5,
use_glow=False)

            # Add metrics score
            add_text_with_style(combined, f"Score: {metrics_data['overall_score']:.2f}", (w-180, h-20),
0.5, use_glow=False)

            # Update the image widget
            update_success = update_image_widget(combined)

            # FPS calculation for camera
            frame_count += 1
            if update_success:
                display_count += 1

            # Update FPS every second
            current_time = time.time()
            elapsed = current_time - start_time
            display_elapsed = current_time - display_time

            if elapsed >= 1.0:
                camera_fps = frame_count / elapsed
                if display_elapsed >= 1.0:
                    display_fps = display_count / display_elapsed
                    status_label.value = f"Mode: {current_processing_mode} | FPS: {camera_fps:.1f} |
Score: {metrics_data['overall_score']:.2f}"
                    display_count = 0
                    display_time = current_time
                else:
                    status_label.value = f"Mode: {current_processing_mode} | FPS: {camera_fps:.1f} |
Score: {metrics_data['overall_score']:.2f}"

                frame_count = 0
                start_time = current_time

            # Short sleep to prevent CPU overuse
            time.sleep(0.01)

        except Exception as e:
            with debug_output:
                print(f"Error in camera processing: {str(e)}")
                import traceback
                traceback.print_exc()

        finally:
            # Clean up
            with debug_output:
                print("Stopping camera...")

            try:
                cap.release()
            except Exception as e:
                with debug_output:
                    print(f"Error releasing camera: {str(e)}")
```

```
        # Take a final snapshot for display in the GUI
        if 'combined' in locals():
            update_image_widget(combined)

        # Reset GUI
        start_button.disabled = False
        stop_button.disabled = True
        mode_dropdown.disabled = False
        current_processing_mode = None
        status_label.value = "Camera stopped"

        with debug_output:
            print("Camera stopped and resources released")

# Start button handler
def on_start_button_click(b):
    global camera_running, current_processing_mode

    if camera_running:
        return

    # Clear output and reset state
    with debug_output:
        clear_output(wait=True)
        print(f"Starting camera in {mode_dropdown.value} mode...")

    # Update GUI
    start_button.disabled = True
    stop_button.disabled = False
    mode_dropdown.disabled = False  # Allow changing modes while running
    status_label.value = "Starting camera..."

    # Initialize mode
    current_processing_mode = mode_dropdown.value

    # Start camera
    camera_running = True

    # Run in a separate thread to keep UI responsive
    camera_thread = threading.Thread(target=process_camera)
    camera_thread.daemon = True
    camera_thread.start()

# Stop button handler
def on_stop_button_click(b):
    global camera_running, current_processing_mode

    with debug_output:
        print("Stopping camera...")

    camera_running = False
    current_processing_mode = None
    status_label.value = "Stopping camera..."

# Show metrics button handler
def on_show_metrics_button_click(b):
    # Display the metrics charts
    display_metrics_charts()
```

```
# Force mode change handler
def on_force_change_button_click(b):
    """Force a mode change for testing"""
    if not camera_running:
        with debug_output:
            print("Camera not running, cannot change mode.")
        return

    # Get current mode index
    options = mode_dropdown.options
    current_idx = options.index(mode_dropdown.value)

    # Get next mode
    next_idx = (current_idx + 1) % len(options)
    next_mode = options[next_idx]

    # Update dropdown
    with debug_output:
        print(f"Forcing mode change: {mode_dropdown.value} -> {next_mode}")

    # Update the dropdown value
    mode_dropdown.value = next_mode

    # Update status
    status_label.value = f"Mode changing to: {next_mode}"

# Attach button handlers
start_button.on_click(on_start_button_click)
stop_button.on_click(on_stop_button_click)
show_metrics_button.on_click(on_show_metrics_button_click)
force_change_button.on_click(on_force_change_button_click)
stop_button.disabled = True

# Create a stylized 80s style predator-themed image for the initial display
mask_img = np.zeros((480, 640*2, 3), dtype=np.uint8)

# Add a dark green glow in the center left
cv2.circle(mask_img, (320, 240), 150, (0, 20, 0), -1)

# Draw stylized predator mask outline
points = np.array([[220, 140], [420, 140], [380, 340], [260, 340]], np.int32)
points = points.reshape((-1, 1, 2))
cv2.polylines(mask_img, [points], True, PREDATOR_GREEN, 1)
cv2.ellipse(mask_img, (320, 200), (50, 30), 0, 0, 360, PREDATOR_GREEN, 1)

# Add targeting reticle in bottom left
cv2.circle(mask_img, (100, 380), 30, PREDATOR_GREEN, 1)
cv2.line(mask_img, (60, 380), (90, 380), PREDATOR_GREEN, 1)
cv2.line(mask_img, (110, 380), (140, 380), PREDATOR_GREEN, 1)
cv2.line(mask_img, (100, 340), (100, 370), PREDATOR_GREEN, 1)
cv2.line(mask_img, (100, 390), (100, 420), PREDATOR_GREEN, 1)

# Add grid lines for the mask
for i in range(180, 350, 40):
    cv2.line(mask_img, (220, i), (420, i), PREDATOR_GREEN, 1)
for i in range(240, 420, 40):
    cv2.line(mask_img, (i, 140), (i, 340), PREDATOR_GREEN, 1)

# Predator's iconic shoulder cannon scope lines
cv2.line(mask_img, (80, 360), (120, 400), (0, 100, 0), 1)
```

```
cv2.line(mask_img, (70, 370), (110, 410), (0, 100, 0), 1)

# Add 80s style predator text to image
font_scale = 1.0
thickness = 2
message = "VANISH SYSTEM READY"
text_size = cv2.getTextSize(message, cv2.FONT_HERSHEY_SIMPLEX, font_scale, thickness)[0]
text_x = (mask_img.shape[1] - text_size[0]) // 2
text_y = 350

# Add glow effect
cv2.putText(mask_img, message, (text_x, text_y),
            cv2.FONT_HERSHEY_SIMPLEX, font_scale, (0, 50, 0), thickness*3)
cv2.putText(mask_img, message, (text_x, text_y),
            cv2.FONT_HERSHEY_SIMPLEX, font_scale, PREDATOR_GREEN, thickness)

# Add start instruction
instruction = "Click 'Start Camera' to begin"
inst_size = cv2.getTextSize(instruction, cv2.FONT_HERSHEY_SIMPLEX, 0.7, 1)[0]
inst_x = (mask_img.shape[1] - inst_size[0]) // 2
inst_y = text_y + 50
cv2.putText(mask_img, instruction, (inst_x, inst_y),
            cv2.FONT_HERSHEY_SIMPLEX, 0.7, PREDATOR_GREEN, 1)

# Use the predator mask as the landing page image
blank_image = mask_img

# Display the initial landing page image
update_image_widget(blank_image)

# GUI layout with custom styling
ui = widgets.VBox([
    # Apply custom CSS
    widgets.HTML(custom_css),
    # Custom title with predator styling
    widgets.HTML("<h2 class='vanish-title'>VANISH CAMOUFLAGE SYSTEM</h2>"),
    widgets.HBox([start_button, stop_button, show_metrics_button, force_change_button]),
    widgets.HBox([mode_dropdown, status_label]),
    image_widget,
    metrics_output,
    debug_output
])

# Display GUI
display(ui)

with debug_output:
    print("VANISH system ready")
    print(f"OpenCV version: {cv2.__version__}")
    print("Click 'Start Camera' to begin")
```

## A.2 VANISH Code Implementation Workflow

- **Build the Network**: Use K-means and a Multi-stage Gaussian blurring pipeline (OpenCV) to develop and implement the model.

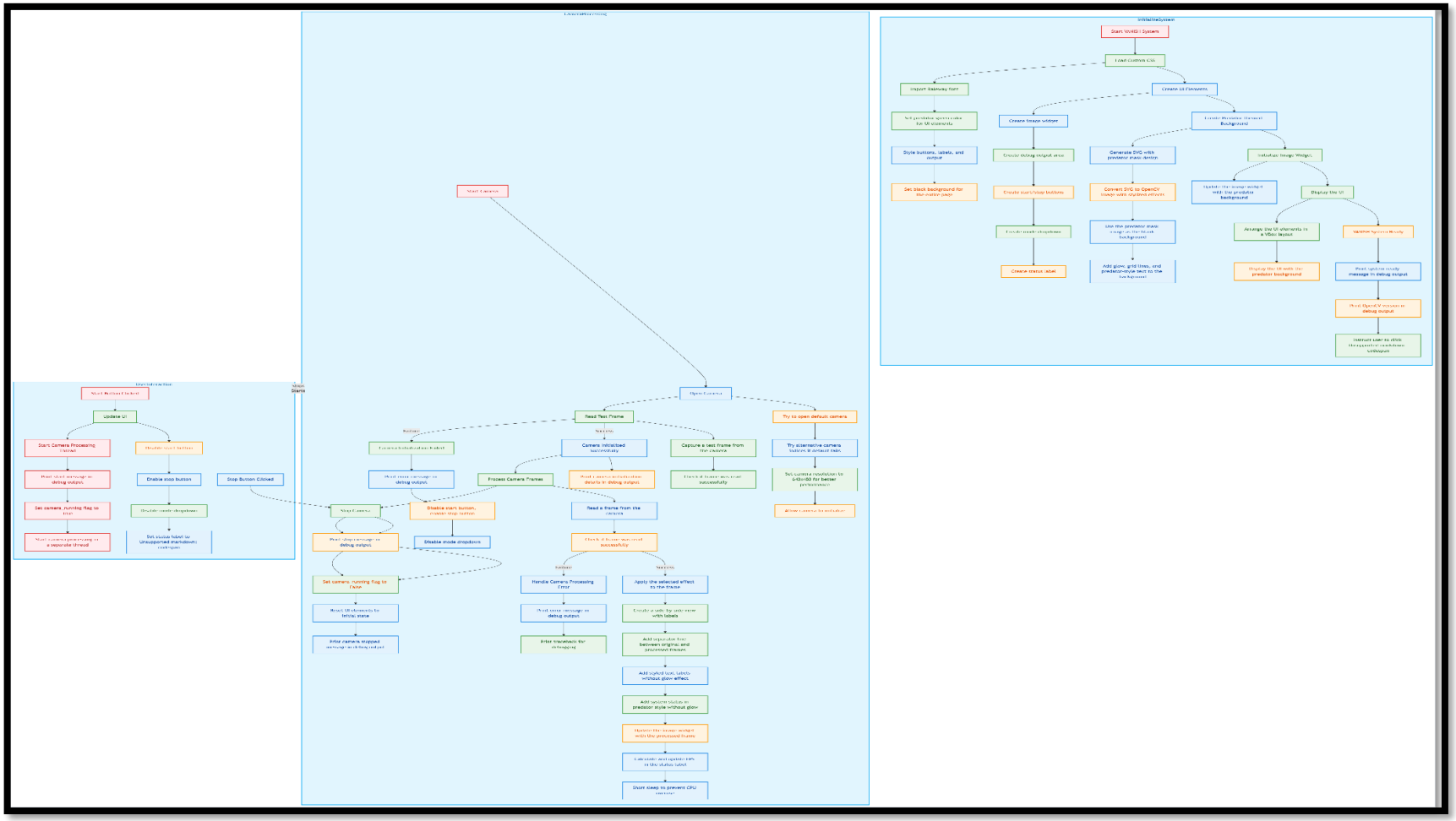


*Figure 51: VANISH Unified Camouflage System Code Framework (https://codetoflow.com/, 2025)*

End of Document