# Deconstruction and Analysis of an Autonomous Security and Interception (ASI) UAV System for Master's Thesis Scoping

## I. Deconstruction of the Autonomous Security & Interception (ASI) Concept

The proposal for an autonomous Unmanned Aerial Vehicle (UAV) capable of independent scanning, threat identification, autonomous takeoff and landing, and kinetic engagement of security risks represents a highly ambitious, system-of-systems challenge. Such a concept integrates multiple, complex research domains from mobile robotics and artificial intelligence.

A formal deconstruction of this vision reveals four primary technical pillars:

- **Pillar 1: Perception (Subsystem I):** This addresses the "find out security thread" requirement. This subsystem involves real-time, on-board object detection, classification, and tracking. These tasks must be performed under the challenging conditions of an aerial viewpoint, where objects are often small, occluded, or viewed from unconventional angles.[1] The system must move beyond simple detection to *contextualize* and identify anomalous or threatening behavior.[3]
- **Pillar 2: Navigation (Subsystem II):** This addresses the "scan all near by area" requirement. In complex environments, such as indoors, in urban canyons, or when GPS is denied, this necessitates Simultaneous Localization and Mapping (SLAM).[4] The drone must be able to construct a map of its unknown surroundings and simultaneously determine its own position within that map.
- **Pillar 3: Autonomous Maneuvering (Subsystem III):** This addresses the "take out and land on the own" requirement. This is a foundational capability covering Autonomous Takeoff and Landing (ATOL), which is critical for system deployment and recovery. It also

includes precision maneuvering, which forms the basis for approaching a designated landing zone or, in a security context, a target of interest.[6]

- **Pillar 4: Kinetic Engagement (Subsystem IV):** This addresses the "attack i mean crash land" requirement. This is the most complex component from both a technical and ethical standpoint. In robotics and aerospace engineering, this is the domain of *terminal guidance*. This phase involves the autonomous control algorithms required to ensure a high-speed, precise impact on a designated target.[8]

The functional description of a drone that can "scan" (loiter), "find" a target (autonomous target recognition), and "attack i mean crash land" (kinetic strike) is a precise match for a technology known in defense and security literature as a "loitering munition" (LM), or colloquially, a "kamikaze drone".[10] Loitering munitions are defined by their ability to "continuously monitoring, locating, and destroying valuable targets".[10] Critically, some LMs are designed to "operate autonomously, searching and destroying targets without any human intervention".[11] This reframing is essential, as it moves the proposal from a generic robotics project into the category of a real-world, dual-use military technology, which is the subject of intense international debate.[11]

Furthermore, the user's two primary constraints—"without any manul control" and "it can attack"—are in direct academic and legal conflict. The combination of full autonomy with a lethal action (a decision to "kill, injure and destroy" [13]) is the precise definition of a Lethal Autonomous Weapon System (LAWS).[14] A system that "select[s] and engage[s] targets" based on its own sensor processing "rather than human inputs" [17] is at the center of an unresolved global ethical and legal debate. The core of the problem is the "accountability gap" [18], as International Humanitarian Law (IHL) is a framework designed for, and addressed to, *human* commanders and operators.[19]

Therefore, a successful Master's thesis must navigate this challenge. An attempt to *literally* build a fully autonomous lethal system would be ethically and legally naive. This analysis will deconstruct each pillar to propose two academically sound paths: 1) Focusing on a single, non-lethal *component* of the system (e.g., perception or navigation) or 2) Reframing the *autonomy* component (e.g., building a *human-supervised* system) to address the complete pipeline in a responsible manner.

## II. Subsystem Analysis I: Real-Time Threat Perception and Anomaly Detection

This subsystem functions as the "eyes" of the ASI drone, tasked with processing sensor data to answer the core question: "What am I looking at, and is it a threat?" The primary research challenges in this domain are achieving high accuracy and real-time inference speeds on power- and weight-constrained embedded hardware, particularly from an aerial perspective.

## Dominant Architectures: The YOLOverse vs. Transformers

The dominant approach in academic literature for UAV-based computer vision is the You Only Look Once (YOLO) family of models. A 2024 systematic literature review found that over 39.5% of analyzed UAV-CV studies employed a YOLO-variant.[1] Its popularity stems from its single-pass architecture, which treats object detection as a regression problem [20], making it exceptionally fast and suitable for real-time applications.

Examples of its application are widespread:

- **Security:** YOLOv8 is used in smart drone surveillance systems to identify "people, some potentially dangerous objects, and fire".[21]
- **Disaster Response:** The YOLOv8-seg model has been optimized for post-earthquake search and rescue (SAR), segmenting 28 critical classes including "structurally compromised buildings," "access points (doors, windows, gaps)," and "rescue personnel".[22]
- **Tracking:** YOLO is integrated into Multi-Object Tracking (MOT) frameworks for UAVs, enabling the system to follow objects of interest over time.[23]

While YOLO is the incumbent, new transformer-based architectures are emerging. Models like **UAV-DETR** [25]and other novel pipelines[2]are being developed specifically to address the key weakness of aerial imagery: detecting "small and occluded objects".[25]Some research claims these newer models can outperform YOLOv8 by a small margin (e.g., 2.6%) on specialized aerial datasets like City-UAV.[26]

## The "Lightweight" Imperative: Deployment on Embedded Systems

A state-of-the-art (SOTA) deep learning model is useless if it is too large or slow to run on the drone itself.[27]A drone's "companion computer" (the on-board AI processor) is a "resource-constrained"[28]embedded system with strict power and thermal limits.[29]

This constraint has created a research field dedicated to creating "lightweight" yet powerful models. A common and highly successful methodology is to integrate optimization modules into a fast baseline. For example, research on real-time weapon detection[27]proposes:

1. Start with a fast baseline (e.g., YOLOv5s).
2. Integrate **GhostNet modules** into the model's head or neck.
3. The GhostNet module reduces the computational load (GFLOPs) and model parameters significantly. One study found a 2.7 reduction in GFLOPs and a 1.42 reduction in parameters.[27]
4. Counter-intuitively, this optimization can also *increase* accuracy, with the lightweight model achieving a higher mean Average Precision (mAP) than the original.[27]

This lightweighting approach is what makes deployment on embedded hardware like the NVIDIA Jetson series feasible.[29]

## Defining the "Security Thread": A Supervised vs. Unsupervised Problem

The concept of a "security thread" is abstract. A deep learning model cannot be trained on "threats"; it must be trained on concrete, observable *proxies* for threats. The literature presents two distinct pathways to solve this.

**Path A (Supervised Object Detection):** This is the most common approach. The "threat" is defined as a specific, pre-trained class of object.

- **Weapons:** A well-established field involves training models to detect "dangerous weapons".[27]
- **Intruders:** Models can be trained on "people" or "person detection" to identify intruders in a restricted area.[21]
- **Anomalous Objects:** The system can be trained to find "fire" or other defined hazards.[21]

**Path B (Unsupervised Anomaly Detection):** This is a more complex but more powerful approach. The "threat" is defined as a *deviation from normal patterns*.[3]This leverages "statistical models, supervised and unsupervised learning techniques, and deep learning algorithms to identify deviations from normal patterns".[3] This allows the drone to flag *unknown* or *unforeseen* threats that were not in its training data, such as a vehicle driving on a pedestrian path or a person exhibiting erratic behavior.[36]

A thesis project must first choose its path: is it building a *better, faster weapon detector* (Path A) or a *novel behavioral anomaly detector* (Path B)?

### Detection vs. Segmentation: Implications for Downstream Tasks

The choice of perception model has a cascading effect on all other subsystems.

1. **Object Detection (e.g., YOLO):** This model draws a *bounding box* around an object.[20]It is fast and answers "What is it?" and "Where is it (approximately)?"
2. **Instance Segmentation (e.g., Mask R-CNN, YOLOv8-seg):** This model creates a pixel-perfect *mask* outlining the object's exact shape.[22]It is slower but provides far richer data.

This choice directly impacts the "attack" and "scan" goals. For the "attack" (Kinetic Engagement) subsystem, a simple bounding box is likely sufficient to provide a target-lock and guide the drone. However, for the "scan area" (Navigation) subsystem, segmentation is vastly superior for *scene understanding*.

The D'RespNeT project proves this.[22] Their YOLOv8-seg model doesn't just find "debris" (a bounding box); it segments *the difference between* "accessible entry points (doors, windows, gaps)" and "obstructed areas." This rich, pixel-level semantic information is what allows a navigation algorithm to make intelligent decisions, far beyond what a simple bounding box could provide. A project focused on pure speed-to-target might choose a lightweight detector, while a project focused on rich navigation and scene understanding must use segmentation.

## III. Subsystem Analysis II: Autonomous Navigation and Semantic Mapping

This subsystem provides the drone with spatial awareness, enabling it to "scan all near by area" and build an internal model of its world. This is paramount in "GPS-denied" environments[38]or any unfamiliar territory.[4]

### Core Technology: Simultaneous Localization and Mapping (SLAM)

The core technology is **SLAM**, an algorithm that allows a robot to simultaneously "build a map of an unfamiliar environment" and "estimate its sensor motion" (i.e., its own location) within that map.[4]

A classic trade-off exists in SLAM sensor modalities:

- **Visual SLAM (vSLAM):** Uses cameras (e.g., ORB-SLAM [4]). It is low-cost, low-weight, and provides dense, feature-rich data. However, it is known to fail in "feature-scarce" (e.g., blank walls, darkness) or "aggressive motion" scenarios.[39]
- **LiDAR SLAM:** Uses LiDAR sensors (e.g., LOAM, F-LOAM [4]). It is "highly effective" and provides "high-precision 3D spatial information," and is robust to lighting conditions.[39]However, it is typically more expensive, heavier, and provides sparser data.
- **Visual-Inertial Odometry (VIO) / Fusion:** This is the modern standard. It fuses vSLAM data (from a camera) with an Inertial Measurement Unit (IMU). The IMU's high-frequency motion data compensates for the camera's weaknesses, allowing "UAVs to navigate challenging environments autonomously and reliably, irrespective of illumination conditions" [38]and aggressive motion.[39]

## Beyond Geometry: Semantic and Open-Vocabulary SLAM

A critical limitation of classic SLAM is that it builds a *geometric* map (a point cloud).[4] This map can answer "Where am I?" and "Where is the wall?" but it *cannot* answer the query, "Where is the *security threat*?"

To solve this, the Navigation subsystem must be *fused* with the Perception subsystem (Section II). This advanced research field is known as **Semantic SLAM**.[38]A Semantic SLAM system creates maps that are both "geometrically accurate and semantically expressive".[44] Instead of a map of "points," the system builds a map of *objects* and *classes*, such as "person," "car," or "door".[42] The user's query implicitly *requires* a Semantic SLAM system to find a "thread" within a "nearby area."

A limitation of traditional Semantic SLAM, however, is that it is "closed-set".[43]It can only identify and map the ~80 classes it was pre-trained on (e.g., from the COCO dataset[1]). It would be blind to a "suspicious backpack" or a "leaking pipe" if not specifically trained on those classes.

The newest and most advanced research, with papers emerging in 2024-2025, solves this with **Open-Vocabulary SLAM**.[43] These systems integrate large Vision-Language Models (VLMs) to create maps that can be queried with *natural language*.

Two exemplar frameworks stand out:

1. **VLMaps:** This framework creates "a spatial map representation that directly fuses pretrained visual-language features with a 3D reconstruction".[41] This allows for complex, spatial, text-based commands, such as "go... *in between the sofa and the TV*".[41]
2. **FindAnything:** This is an "open-world mapping and exploration framework" that "incorporates vision-language information into dense volumetric submaps".[44]It allows a robot to explore based on natural language queries and is explicitly noted as "the first of its kind to be deployed on resource-constrained devices, such as MAVs".[44]

This cutting-edge field is an ideal and highly publishable topic for a Master's thesis, directly addressing the need to find abstract "threats" in a scanned environment.

## Scanning Large Areas: Collaborative SLAM (C-SLAM)

For scanning a *large* area, a single drone is inefficient. The solution is a "swarm" or multi-drone system.[23] **Collaborative SLAM (C-SLAM)** [47]is a technique where multiple UAVs "build a collective representation of the environment" and "shar[e] situational awareness".[47]This involves complex distributed systems challenges, such as merging maps and performing "inter-drone loop closures"[49]to create a single, unified world model.[50]

## Table 1: Comparative Analysis of SLAM Modalities for UAV Navigation

| Modality | Key Algorithms | Sensors Used | Pros | Cons |
|---|---|---|---|---|
| **vSLAM** | ORB-SLAM[4] | Monocular or Stereo Camera | Low cost, low weight, dense data | Fails in low-light/low-texture[39], |

| | | | | |
|---|---|---|---|---|
| | | | | sensitive to motion |
| **LiDAR SLAM** | LOAM, F-LOAM[4] | 2D/3D LiDAR | High precision, robust to lighting [40] | Expensive, heavy, sparse data, struggles with fast rotation |
| **VIO/Fusion** | (e.g.[38]) | Camera + IMU | **Robust SOTA**; handles aggressive motion & textureless areas [38, 39] | Increased computational complexity |
| **Semantic SLAM** | (e.g.[42]) | Camera/LiDAR + DL Detector (YOLO) | Creates object-level maps ("person," "car")[42] | "Closed-set": only finds pre-trained classes[43] |
| **Open-Vocab SLAM** | VLMaps[41], FindAnything[44] | Camera + VLM (e.g., CLIP) | **Cutting-Edge**; "open-world" [44], understands text queries[41] | High computational cost, research is very new |

## IV. Subsystem Analysis III: Precision Autonomous Takeoff, Landing, and Target Approach

This subsystem covers the drone's autonomous maneuvering for both utility ("take out and land on the own"[7]) and the non-terminal phase of target approach. The core technology for both applications is **Visual Servoing (VS)**.

## Core Technology: Visual Servoing (VS)

Visual Servoing is a control technique that uses feedback from a vision sensor (camera) to control the motion of a robot.[52]Rather than relying on an abstract world coordinate, it "calculat[es] velocity commands directly in image space"[7]to, for example, keep a target centered in its view.

- **Image-Based Visual Servoing (IBVS):** Performs control calculations based on 2D features in the image plane. It is fast and robust to camera calibration errors.[52]
- **Position-Based Visual Servoing (PBVS):** Reconstructs the full 3D pose of the target relative to the camera, then performs control in 3D space. It can be more stable but is highly dependent on accurate calibration and 3D estimation.[52]

A robust VS system for a quadrotor typically relies on a "minimum sensor set" consisting of an Inertial Measurement Unit (IMU), an ultrasonic sensor (for altitude), and a vision sensor.[52]

## Application 1: Autonomous Landing (ATOL)

This capability can be divided into two distinct problems:

1. **Static Targets:** This is the "baseline" problem. It involves landing on a known, static landing pad.[54] This is often solved by using *fiducial markers* (like AprilTags [55]) that are easily and robustly detected by the CV system.[56]
2. **Dynamic Targets (Moving Platforms):** This is a much more complex and academically rich problem, with numerous papers dedicated to landing a UAV on a moving vehicle or boat.[6]

The problem of landing on a moving platform serves as an excellent, self-contained Master's thesis. It requires a complete robotics pipeline:

- **Detection:** The drone must detect the moving platform (e.g., a car roof) using its on-board monocular camera.[6] Modern approaches use YOLO for this task.[59]
- **State Estimation:** The drone cannot just see the platform; it must *predict* its future movement using a "nonlinear motion model" [6] or an Extended Kalman Filter (EKF).[55]
- **Control:** A robust visual servoing controller [7] or an advanced "boundary layer sliding controller" [55] is used to follow the target and execute the landing maneuver.

Crucially, this project serves as a *non-lethal proxy* for the "attack" goal. The control problem of "landing on" a moving target is dynamically similar to "crashing on" it. Both require high-speed, precision tracking and a controlled maneuver to a terminal point, with the only major difference being the desired terminal velocity (zero for landing, high for impact).

# V. Subsystem Analysis IV: Terminal Guidance and Kinetic Neutralization

This section provides a technical analysis of the "attack i mean crash land" request. This is not landing; it is *interception*. The objective is to achieve a *hard impact* with a target, which may be static or, in the more challenging case, non-cooperative and maneuvering.

## Reframing: From "Crash" to "Terminal Guidance"

This problem is known in academic literature as *terminal guidance* or *interception*.[8] This is an active and legitimate field of engineering research, with papers regularly accepted to top-tier robotics conferences like ICRA (International Conference on Robotics and Automation) and IROS (International Conference on Intelligent Robots and Systems).

The technical literature is explicit about this capability:

- An **IROS 2025** paper describes a strategy where, "Upon breaching a warning threshold, the UAVs may even employ a **suicide attack to neutralize the hostile target**".[8]
- An **ICRA** paper proposes a scheme for an "autonomous multicopter... to intercept a maneuvering intruder UAV," which was validated by achieving a "high-speed interception... with a **terminal speed of 20 m/s**".[9]
- Another paper aims for "precise interception" at the "centimeter level".[60]

This confirms that the "crash" concept is a valid and cutting-edge *control systems* problem. A technical thesis can focus purely on the mathematics, dynamics, and control theory of interception, typically validated in simulation.

## The Control Law: Proportional Navigation (PN)

A key distinction must be made. A landing controller (Section IV) is designed to drive the relative velocity between the drone and target to *zero*.[7] An interception controller uses a *guidance law* to *guarantee collision*.

The classic guidance law, borrowed from decades of missile development, is **Proportional Navigation (PN)**.[60] In its simplest form, PN guidance states that the interceptor's turning rate should be directly proportional to the rate of change of the line-of-sight (the angular speed) to the target.

Modern robotics research *fuses* this classic missile guidance law with modern computer vision control:

- One paper designs an "Image-Based Visual Servoing (IBVS) control algorithm **based on proportional navigation guidance**".[60]
- Another discusses "Terminal attack trajectories" and their relationship to the "proportional navigation guidance law".[62]

A strong, technical Master's project would be the "Implementation of a Proportional Navigation Guidance Law using Image-Based Visual Servoing for UAV Interception." This project would focus on the high-speed dynamics and control theory [62], using the CV pipeline [9] as the sensor input.

### The Target: Static vs. Maneuvering

The complexity of this project is determined by the target.

- **Static Ground Target:** This is the simpler problem, analogous to a precision landing but with a high terminal velocity.[54]
- **Dynamic/Maneuvering Target:** This is the more complex and academically novel challenge, such as intercepting another non-cooperative drone.[8] This requires advanced state estimation and trajectory prediction of the target [61], making it a more significant research undertaking.

## VI. Critical Analysis: Human Control and the Lethal Autonomy Problem

Any academic work on this topic must include a non-negotiable component: a critical analysis of the "without any manul control" requirement when combined with the "attack" function.

This forms the ethical and legal foundation of the thesis.

## The Core Problem: Lethal Autonomous Weapons Systems (LAWS)

As established, the proposed system—a drone that "select[s] and engage[s] targets based on sensor processing, rather than human inputs" [17]—is a LAWS.[14] Such systems raise profound ethical objections about "delegat[ing] decisions to kill, injure and destroy... to machines".[13]

## The Legal Framework: International Humanitarian Law (IHL)

Any weapon system, autonomous or not, must be capable of being used in compliance with IHL.[63] The core principles of IHL, which are addressed to *human commanders*, are [19]:

1. **Distinction:** The ability to distinguish between combatants and civilians.
2. **Proportionality:** The judgment that incidental civilian harm (collateral damage) is not excessive in relation to the anticipated military advantage.
3. **Precaution:** Taking all feasible precautions to avoid civilian harm.

A fully autonomous system poses a direct challenge to this legal framework. If a drone, acting on its own, misidentifies a target (e.g., a farmer for a soldier, a risk noted in [12]) and executes a lethal strike, it creates what legal experts call the "accountability gap".[18] IHL rules "are addressed to humans" [19], and it is humans who are held accountable for violations. The ICRC and others have stated that legal accountability "cannot be transferred to a machine, computer program or weapon system".[19]

## Case Study: The 2021 UN Report on the STM Kargu-2 in Libya

This is not a theoretical problem. In 2021, a UN Panel of Experts report on the conflict in Libya [63] became the "ground zero" for the LAWS debate.[67] The report stated that a Turkish-made STM Kargu-2 loitering munition was potentially used "to attack targets without requiring data connectivity between the operator and the munition: in effect, a true 'fire, forget and find' capability".[63] This "would mark the first publicly-known instance of an AI-equipped weapons system autonomously identifying and engaging combatants".[67] This real-world incident perfectly illustrates the system in question and the intense legal and ethical scrutiny that immediately follows.[66]

## An Academically Defensible Alternative: Human-on-the-Loop (HOTL)

The "no manual control" concept is one end of a spectrum. The literature provides a more nuanced and practical taxonomy of human control.[16]

1. **Human-in-the-Loop (HITL):** The machine finds a target and proposes an action, but it *must stop* and *wait* for explicit human permission to "fire".[15] This is safe but can be too slow for time-critical threats.
2. **Fully Autonomous (LAWS):** The machine performs the entire "detect-to-engage" sequence without human intervention.[17] This is the legally and ethically problematic model.[13]
3. **Human-on-the-Loop (HOTL) / Supervised Autonomy:** This is the practical, ethically-defensible middle ground.[69] The machine is *fully autonomous* in its sensing, planning, and *even its decision to engage*... **but** a human operator "monitor[s] the AWS at all times and can intervene when required".[69] The machine acts *unless* the human operator actively vetoes the action.

This HOTL model provides a robust framework for a Master's thesis. The technical challenge remains: build a fully autonomous system. But the *research contribution* is in designing the "fail-safe" or "veto" mechanism that ensures "Meaningful Human Control" (MHC).[69]

The canonical example of this is the U.S. C-RAM (Counter-Rocket, Artillery, and Mortar) system.[71] The C-RAM "autonomously" identifies, tracks, and aims at incoming rockets. The human operator does *not* aim. Their role is "to act as a final fail-safe" and press the button authorizing engagement.[71] This "centaur human-machine teaming" [71] provides a rigorous and responsible academic framework for the project.

## Table 2: Comparative Analysis of Human Control Models for LAWS

| Control Model | Definition | Operator Role | Example System | Ethical/IHL Compliance |
| --- | --- | --- | --- | --- |

| | | | | |
|---|---|---|---|---|
| **Human-in-the-Loop (HITL)** | Machine selects target, but **must** get human permission to engage. | **Active Approver.** Final "fire" decision is human. [15] | Semi-autonomous guided missiles. | Generally compliant. Human retains control over lethal decision. |
| **Human-on-the-Loop (HOTL)** | Machine selects *and* engages targets autonomously. | **Active Supervisor (Veto).** Human monitors and can *intervene* or *cancel* the attack. [69] | Counter-Rocket (C-RAM) systems.[71] | **The "Grey Area."** Argued as compliant if the human has "Meaningful Human Control" (MHC).[69] |
| **Fully Autonomous (LAWS)** | Machine selects and engages targets without *any* human in the decision loop. | **Passive Monitor (or no human).** "Fire, forget and find." [17] | STM Kargu-2 (alleged).[63] | **Legally Contested.** Raises "accountability gap" [18] and ethical concerns.[13] |

## VII. Proposed Master's Thesis Project Scopes (A Synthesis)

Based on the subsystem analysis, the original concept can be scoped into several distinct, feasible, and academically rigorous Master's-level projects.

### Project 1: Embedded Real-Time Threat Detection for UAVs (Focus: Subsystem I)

- **Title:** "A Lightweight Deep Learning Framework for Real-Time Weapon and Personnel Detection on Resource-Constrained UAVs."

- **Objective:** To design, train, and deploy a highly efficient object detection model on an embedded (on-board) computer for a UAV.
- **Methodology:**
  1. **Baseline:** Implement a SOTA model (e.g., YOLOv8n [33]).
  2. **Contribution:** Create a "lightweight" variant by integrating optimization modules like GhostNet [27] or applying quantization/pruning.
  3. **Dataset:** Train on a specialized security dataset (e.g., Sohas weapon dataset [27]) or a disaster/anomaly dataset (D'RespNeT [22]).
  4. **Deployment:** Deploy the model on an NVIDIA Jetson platform [29] and benchmark its performance in mAP (accuracy), GFLOPs (computation), and FPS (speed).
- **Addresses:** "find out security thread automatically" + real-world hardware constraints.

## Project 2: Open-Vocabulary Semantic SLAM for Security Surveillance (Focus: Subsystem II)

- **Title:** "Open-World Scene Understanding for Autonomous UAVs: An Open-Vocabulary Semantic SLAM Framework."
- **Objective:** To build a navigation system that allows a UAV to create a 3D map of an unknown environment and identify arbitrary objects or "threats" using natural language queries.
- **Methodology:**
  1. **SLAM:** Implement a robust VIO or vSLAM pipeline (e.g., ORB-SLAM3 [72]) within ROS 2.
  2. **Contribution:** Integrate a Vision-Language Model (VLM) to create an "open-vocabulary" map, based on SOTA research like **VLMaps** [41] or **FindAnything**.[44]
  3. **Deliverable:** A system where the drone autonomously explores an area (in simulation or a lab), builds a 3D map, and a human operator can then query it: "Show me all *laptops*," "Where is the *backpack*?", "Highlight *structural gaps*".[41]
- **Addresses:** "scan all near by area" + "find... thread" in a novel, non-lethal, and highly advanced way.

## Project 3: Vision-Based Terminal Guidance for Non-Cooperative Target Interception (Focus: Subsystem IV - Simulation)

- **Title:** "Terminal Guidance of a UAV Interceptor using Proportional Navigation and Image-Based Visual Servoing."
- **Objective:** To design and validate *in simulation* a high-speed control system that guides

a UAV to intercept (i.e., "crash land on") a non-cooperative *moving* target.

- **Methodology:**
  1. **Simulation:** Build a high-fidelity model of a quadrotor and a target in Gazebo [73] or AirSim.[74]
  2. **Perception:** Use a simple CV-based tracker (e.g., OpenCV [75]) to get the target's 2D image-plane coordinates.
  3. **Contribution:** Design and implement a controller that fuses **Image-Based Visual Servoing (IBVS)** with the **Proportional Navigation (PN) guidance law**.[60]
  4. **Validation:** Replicate the results of [9], demonstrating high-speed interception (e.g., 20 m/s) with centimeter-level accuracy [60] on a maneuvering target.
- **Addresses:** "attack i mean crash land" in a *purely technical, control-theory context*, scoping out the ethical and detection problems to focus on the dynamics.

## Project 4: A Human-on-the-Loop (HOTL) Fail-Safe Framework for Autonomous Systems (Focus: Subsystem VI)

- **Title:** "A Human-on-the-Loop Architecture for Veto-Based Supervisory Control of an Autonomous Security UAV."
- **Objective:** To design and simulate a robust HOTL control framework that *guarantees* a human operator can safely monitor and *veto* the actions of an autonomous drone performing target selection.
- **Methodology:**
  1. **Autonomous Core:** Simulate a drone autonomously performing a task (e.S., detecting and tracking a target using YOLO [21]).
  2. **Interface:** Create a ground-station interface that displays the drone's video feed, its *intended action* (e.g., "Target Locked - Engaging"), and a large "VETO" button.
  3. **Contribution:** Design the software architecture (using ROS 2) that manages the "supervised autonomy".[69] The research challenge is ensuring the veto command is respected, handling latency, and defining the "Rules of Engagement" [65] that allow the human to act as a "final fail-safe".[71]
- **Addresses:** The *entire* user query, but reframed in an ethically and academically responsible manner.

## Project 5: The "Full-Stack LITE" (Civilian Application) (Focus: Subsystems I, II, III)

- **Title:** "An Autonomous UAV System for Post-Disaster Infrastructure Inspection: Fusing Semantic SLAM and Precision Landing."
- **Objective:** To build and demonstrate a *non-lethal* version of the user's entire pipeline, replacing the "security threat" with a "structural anomaly."
- **Methodology:**
  1. **ATOL:** Implement autonomous takeoff.[77]
  2. **Navigation:** Use Semantic SLAM [38] to "scan... area" (e.g., a building facade or bridge).
  3. **Perception:** Use a model like YOLOv8-seg [22] to "find... thread," redefined as "cracks," "debris," or "structural gaps".[1]
  4. **Maneuvering:** Instead of "attacking," use precision visual servoing [52] to autonomously *approach* a detected anomaly for close-up inspection, then "land on the own" [54] at its base.
- **Addresses:** All components of the user's query in a cohesive, ambitious, and non-controversial civilian application.

# VIII. Implementation Framework: A Recommended Hardware/Software Stack

This section provides a practical, technical roadmap for implementing any of the projects above, based on common academic and open-source standards.

## Hardware Selection

- **Flight Controller: Pixhawk Series (e.g., Pixhawk 4, 5x, 6x)**
  - **Rationale:** This is the *de facto* open-source standard for autonomous vehicles. It runs the PX4 Autopilot [78] or Ardupilot. Its sole purpose is to handle all low-level, real-time flight control, stabilization, and safety features.[79] This offloads the critical task of "not crashing" from the high-level AI computer.
- **Companion Computer: NVIDIA Jetson Series**
  - **Rationale:** The project is defined by Deep Learning-based Computer Vision, which requires a powerful GPU for parallel processing. A Raspberry Pi [80] is insufficient for real-time SOTA models, and off-board processing introduces disqualifying latency. The Jetson line is designed for on-board, real-time AI inference at the "edge".[28]
  - The choice of Jetson is a primary design constraint that dictates the feasible

complexity of the DL models.

## Table 3: Performance Benchmarks of Embedded Platforms for YOLO Inference

(Data synthesized from [29])

| Platform | AI Performance (TOPS) | GPU Architecture | Power | Est. Cost | Real-Time YOLO Capability |
|---|---|---|---|---|---|
| **Jetson Nano** | 0.472 (472 GFLOPS) | 128-core Maxwell | 5-10W | Low | **Marginal.** [31] (11 FPS on PeopleNet). Struggles with SOTA models. |
| **Jetson Xavier NX** | 21 TOPS | 384-core Volta | 10-15W | Medium | **Excellent.** "Does better than... Nano".[29] Can run multiple complex models.[30] |
| **Jetson Orin Nano 8GB** | 40 TOPS | 1024-core Ampere | 7-15W | Medium-High | **SOTA.** The modern successor to Xavier. Best performance-per-watt. |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | [81, 82] |

- **Recommendation:** For a Master's thesis, the **Jetson Orin Nano 8GB** [82] or **Xavier NX 16GB** [82] is the correct choice. The Jetson Nano [29] is now significantly underpowered for SOTA computer vision research.
- **Sensors: Intel RealSense D435i or D455**
  - **Rationale:** This is an ideal all-in-one sensor package. It provides: 1) An **RGB Camera** for all CV tasks [75]; 2) A **Stereo Depth Camera** for 3D mapping and SLAM [38]; and 3) An **IMU** for VIO and stabilization.[72] This single sensor provides all necessary inputs for Project 2 (Semantic SLAM) or Project 5 (Full-Stack LITE).

## Software Architecture

- **OS/Middleware: Ubuntu 20.04/22.04 with ROS 2 (Humble/Iron)**
  - **Rationale:** The Robot Operating System (ROS) is the universal standard for robotics development.[75] It provides a "modular software architecture" [79] and a vast ecosystem of pre-built packages for SLAM [72], CV [75], and control.[85] ROS 1 (Noetic) is end-of-life [86], making ROS 2 the mandatory choice for new projects. ROS 2 also offers "a much deeper and lower-latency integration with PX4".[78]
- **Flight Control Interface: MAVROS or XRCE-DDS**
  - A "bridge" is required to communicate between ROS 2 (on the Jetson, making high-level decisions like "go to target") and PX4 (on the Pixhawk, executing low-level motor commands).
  - **MAVROS:** The traditional, well-supported bridge. It "translat[es] ROS messages into MAVLink commands".[79]
  - **XRCE-DDS:** The *newer, faster*, ROS 2-native bridge. The PX4 development team *highly recommends* this, as it "allows direct access to PX4 from ROS 2 workflows" and is the superior choice for "low latency... control".[78] For a high-speed project like Project 3 (Interception), this is the correct technical choice.
- **Simulation Environment: Gazebo vs. AirSim**
  - All development *must* begin in simulation.[77] The choice of simulator depends on the project's bottleneck.
  - **Gazebo:** The ROS-standard simulator.
    - **Pros:** "Larger community" [88], robust physics simulation [90], and seamless ROS integration.[73] It can simulate "sensors, including LIDAR, cameras".[91]
    - **Cons:** Simplistic, non-photorealistic visuals.
  - **AirSim:** A high-fidelity simulator from Microsoft.[74]
    - **Pros:** Uses **Unreal Engine** [88] to provide *photorealistic* visual data, which is "high-fidelity visual... simulation".[74]

        - **Cons:** Less community support and "not well supported as gazebo on ROS / Linux".[88]
    - **Recommendation:**
        - For projects focused on **SLAM, control, or C-SLAM** (Projects 2, 3, 4), use **Gazebo**.[73] The physics and ROS integration are more important.
        - For projects focused *heavily* on the **CV/DL model** (Projects 1, 5), use **AirSim**.[74] The photorealistic camera feed is critical for training a vision model that will transfer to the real world ("sim-to-real").
- **Key Libraries and Stacks:**
    - **OpenCV:** For all fundamental image processing, filtering, and feature tracking.[75]
    - **PyTorch / TensorFlow:** For developing, training, and running the deep learning models.[93]
    - **Open-Source Frameworks:** A Master's student should not start from scratch. Build upon an existing, open-source autonomous UAV stack, such as the **CERLAB-UAV-Autonomy** framework from CMU (C++/ROS-based) [94] or the **MRS UAV System** from CTU.[86] GitHub repositories like [93] provide a direct example of a stack combining "Autonomous-DIH-Drone," "opencv," "deep-neural-networks," "airsim," "pytorch-cnn," "ros-noetic," "pixhawk-flight-controller," and "yolov8-segmentation."

### Referenzen


1. (PDF) Autonomous UAV Navigation using Deep Learning-Based ..., Zugriff am November 4, 2025, https://www.researchgate.net/publication/383601905_Autonomous_UAV_Navigation_using_Deep_Learning-Based_Computer_Vision_Frameworks_A_Systematic_Literature_Review
2. UAV: An Edge-based Energy-Efficient Object Detection System for Unmanned Aerial Vehicles - arXiv, Zugriff am November 4, 2025, https://arxiv.org/html/2308.04774v2
3. (PDF) AI-powered surveillance systems and anomaly detection - ResearchGate, Zugriff am November 4, 2025, https://www.researchgate.net/publication/387933428_AI-powered_surveillance_systems_and_anomaly_detection
4. Comparison of Visual and LiDAR SLAM Algorithms using NASA Flight Test Data, Zugriff am November 4, 2025, https://ntrs.nasa.gov/api/citations/20220017949/downloads/Comparison_of_Visual_and_LiDAR_SLAM_Algorithms_using_NASA_Flight_Test_Data_FINAL.pdf
5. What Is SLAM (Simultaneous Localization and Mapping)? - MATLAB & Simulink, Zugriff am November 4, 2025, https://www.mathworks.com/discovery/slam.html
6. Autonomous Drones Landing On Moving Vehicles - Fly4Future, Zugriff am November 4, 2025, https://fly4future.com/custom-drones/autonomous-drones-landing-on-moving-v

ehicles/
7. Visual Servoing Approach to Autonomous UAV Landing on a Moving Vehicle - PMC - NIH, Zugriff am November 4, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC9459808/
8. Autonomous 3D Moving Target Encirclement and Interception with ..., Zugriff am November 4, 2025, https://www.arxiv.org/abs/2506.13106
9. High-Speed Interception Multicopter Control by Image-based Visual Servoing - arXiv, Zugriff am November 4, 2025, https://arxiv.org/html/2404.08296v1
10. Loitering Munitions: The Convergence of AI, Autonomy, and Lethal Precision in Future Combat by 2029 - MarketsandMarkets, Zugriff am November 4, 2025, https://www.marketsandmarkets.com/blog/AD/loitering-munitions-convergence-ai-autonomy-lethal-precision-future-combat
11. Loitering Munitions 101: What They Are and Why They Matter - IDGA.org, Zugriff am November 4, 2025, https://www.idga.org/command-and-control/articles/loitering-munitions-101-what-they-are-why-they-matter
12. Loitering munitions preview the autonomous future of warfare ..., Zugriff am November 4, 2025, https://www.brookings.edu/articles/loitering-munitions-preview-the-autonomous-future-of-warfare/
13. Ethics and autonomous weapon systems: An ethical basis for human control? - ICRC, Zugriff am November 4, 2025, https://www.icrc.org/en/download/file/69961/icrc_ethics_and_autonomous_weapon_systems_report_3_april_2018.pdf
14. Banning Autonomous Weapons: A Legal and Ethical Mandate | Ethics & International Affairs, Zugriff am November 4, 2025, https://www.cambridge.org/core/journals/ethics-and-international-affairs/article/banning-autonomous-weapons-a-legal-and-ethical-mandate/5FD01B5A96116766C3B1273490B24897
15. Lethal Autonomous Weapon Systems (LAWS): Accountability, Collateral Damage, and the Inadequacies of International Law - Temple iLIT, Zugriff am November 4, 2025, https://law.temple.edu/ilit/lethal-autonomous-weapon-systems-laws-accountability-collateral-damage-and-the-inadequacies-of-international-law/
16. Lethal autonomous weapons systems & artificial intelligence: Trends, challenges, and policies, Zugriff am November 4, 2025, https://sciencepolicyreview.org/wp-content/uploads/securepdfs/2022/08/MITSPR-v3-191618003019.pdf
17. A Hazard to Human Rights: Autonomous Weapons Systems and Digital Decision-Making, Zugriff am November 4, 2025, https://www.hrw.org/report/2025/04/28/hazard-human-rights/autonomous-weapons-systems-and-digital-decision-making
18. Autonomous Weapons Systems and the Need to Update International Humanitarian Law?, Zugriff am November 4, 2025, https://scholarlycommons.law.case.edu/cgi/viewcontent.cgi?article=2702&context

=jil
19. A legal perspective: Autonomous weapon systems under ... - ICRC, Zugriff am November 4, 2025, https://www.icrc.org/sites/default/files/document/file_list/autonomous_weapon_systems_under_international_humanitarian_law.pdf
20. What is the *actual* difference between YOLO and R-CNN models? - Reddit, Zugriff am November 4, 2025, https://www.reddit.com/r/computervision/comments/l5ybwv/what_is_the_actual_difference_between_yolo_and/
21. Smart Drone Surveillance System Based on AI and on IoT Communication in Case of Intrusion and Fire Accident - MDPI, Zugriff am November 4, 2025, https://www.mdpi.com/2504-446X/7/12/694
22. D'RespNeT: A UAV Dataset and YOLOv8-DRN Model for ... - arXiv, Zugriff am November 4, 2025, https://arxiv.org/pdf/2508.16016
23. RAMOTS: A Real-Time System for Aerial Multi-Object Tracking based on Deep Learning and Big Data Technology - arXiv, Zugriff am November 4, 2025, https://arxiv.org/html/2502.03760v1
24. Lightweight Model for Small Drone Detection in Multiple Scenarios - IAENG, Zugriff am November 4, 2025, https://www.iaeng.org/IJCS/issues_v52/issue_9/IJCS_52_9_39.pdf
25. UAV-DETR: Efficient End-to-End Object Detection for Unmanned Aerial Vehicle Imagery, Zugriff am November 4, 2025, https://arxiv.org/html/2501.01855v1
26. Vision-Based Anti Unmanned Aerial Technology: Opportunities and Challenges - arXiv, Zugriff am November 4, 2025, https://arxiv.org/html/2507.10006v1
27. (PDF) A lightweight YOLOv5 for real-time dangerous weapons ...., Zugriff am November 4, 2025, https://www.researchgate.net/publication/381072298_A_lightweight_YOLOv5_for_real-time_dangerous_weapons_detection
28. Benchmarking Deep Learning Models for Object Detection on Edge Computing Devices, Zugriff am November 4, 2025, https://arxiv.org/html/2409.16808v1
29. Zugriff am November 4, 2025, https://www.researchgate.net/figure/Benchmarking-Jetson-Nano-Jetson-Xavier-NX-and-RPi-with-NCS2-Note-Video1-has-1596_tbl3_359689579#:~:text=The%20Xavier%20NX%20does%20better,models%20%5B24%2C%2025%5D.
30. A Review of Embedded Machine Learning Based on Hardware, Application, and Sensing Scheme - PubMed Central, Zugriff am November 4, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC9959746/
31. NVIDIA Jetson Nano and Jetson Xavier NX Comparison: Specifications, Benchmarking, Container Demos, and Custom Model Inference - Latest News from Seeed Studio, Zugriff am November 4, 2025, https://www.seeedstudio.com/blog/2020/06/04/nvidia-jetson-nano-and-jetson-xavier-nx-comparison-specifications-benchmarking-container-demos-and-custom-model-inference/
32. Development and Optimization of Deep Learning Models for Weapon Detection in Surveillance Videos - MDPI, Zugriff am November 4, 2025,

https://www.mdpi.com/2076-3417/12/12/5772
33. IEEE Weapon Detection Paper | PDF | Automation | System - Scribd, Zugriff am November 4, 2025, https://www.scribd.com/document/854955023/IEEE-Weapon-Detection-Paper
34. 9 Best Object Detection Models of 2025: Reviewed & Compared - Hitech BPO, Zugriff am November 4, 2025, https://www.hitechbpo.com/blog/top-object-detection-models.php
35. Deep Learning-Based Anomaly Detection in Video Surveillance: A Survey - MDPI, Zugriff am November 4, 2025, https://www.mdpi.com/1424-8220/23/11/5024
36. End-to-End Deep One-Class Learning for Anomaly Detection in UAV Video Stream - NIH, Zugriff am November 4, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC8321331/
37. A Comparative Analysis Of SSD, Mask R-CNN And Yolo For Object Detection - TIJER, Zugriff am November 4, 2025, https://tijer.org/tijer/papers/TIJERC001183.pdf
38. A semantic SLAM-based method for navigation and landing of UAVs in indoor environments, Zugriff am November 4, 2025, https://www.researchgate.net/publication/379255102_A_semantic_SLAM-based_method_for_navigation_and_landing_of_UAVs_in_indoor_environments
39. A Review of Visual-LiDAR Fusion based Simultaneous Localization and Mapping - PMC, Zugriff am November 4, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC7181037/
40. A Review of Research on SLAM Technology Based on the Fusion of LiDAR and Vision, Zugriff am November 4, 2025, https://www.mdpi.com/1424-8220/25/5/1447
41. Visual Language Maps for Robot Navigation - Autonome Intelligente ..., Zugriff am November 4, 2025, http://ais.informatik.uni-freiburg.de/publications/papers/huang23icra.pdf
42. Visual SLAM for Unmanned Aerial Vehicles: Localization and Perception - MDPI, Zugriff am November 4, 2025, https://www.mdpi.com/1424-8220/24/10/2980
43. A Frontier Review of Semantic SLAM Technologies Applied to the Open World - MDPI, Zugriff am November 4, 2025, https://www.mdpi.com/1424-8220/25/16/4994
44. FindAnything: Open-Vocabulary and Object-Centric Mapping for Robot Exploration in Any Environment - arXiv, Zugriff am November 4, 2025, https://arxiv.org/html/2504.08603v2
45. Dynamic Open Vocabulary Enhanced Safe-landing with Intelligence (DOVESEI) - arXiv, Zugriff am November 4, 2025, https://arxiv.org/html/2308.11471v5/
46. Mixed Human-UAV Reinforcement Learning: Literature Review and Open Challenges - EVENTS @ ISAE-SUPAERO (Indico), Zugriff am November 4, 2025, https://events.isae-supaero.fr/event/14/contributions/367/attachments/32/69/MixedHuman-UAVReinforcementLearningLiteratureReviewandOpenChallenges.pdf
47. Simultaneous Localization and Mapping (SLAM) and Data Fusion in Unmanned Aerial Vehicles: Recent Advances and Challenges - MDPI, Zugriff am November 4, 2025, https://www.mdpi.com/2504-446X/6/4/85

48. Towards Collaborative Simultaneous Localization and Mapping: a Survey of the Current Research Landscape - arXiv, Zugriff am November 4, 2025, https://arxiv.org/pdf/2108.08325
49. Ultra-Lightweight Collaborative Mapping for Robot Swarms - arXiv, Zugriff am November 4, 2025, https://arxiv.org/html/2407.03136v2
50. A Distributed Multi-Robot Collaborative SLAM Method Based on Air–Ground Cross-Domain Cooperation - MDPI, Zugriff am November 4, 2025, https://www.mdpi.com/2504-446X/9/7/504
51. Multi-Drone Cooperation for Improved LiDAR-Based Mapping - MDPI, Zugriff am November 4, 2025, https://www.mdpi.com/1424-8220/24/10/3014
52. Robust Visual Servoing Control for Ground Target Tracking of ..., Zugriff am November 4, 2025, https://www.researchgate.net/publication/334058018_Robust_Visual_Servoing_Control_for_Ground_Target_Tracking_of_Quadrotors
53. Toward Image Based Visual Servoing for Aerial Grasping and Perching - Hybrid Robotics, Zugriff am November 4, 2025, https://hybrid-robotics.berkeley.edu/publications/ICRA2014.pdf
54. Practical Challenges in Landing a UAV on a Dynamic Target - arXiv, Zugriff am November 4, 2025, https://arxiv.org/pdf/2209.14465
55. Dynamic Landing of an Autonomous Quadrotor on a Moving Platform in Turbulent Wind Conditions - DSpace@MIT, Zugriff am November 4, 2025, https://dspace.mit.edu/bitstream/handle/1721.1/136726/1909.11071.pdf?sequence=2&isAllowed=y
56. A Precision Drone Landing System using Visual and IR Fiducial Markers and a Multi-Payload Camera - arXiv, Zugriff am November 4, 2025, https://arxiv.org/html/2403.03806v1
57. UAV precise ATOL techniques using UWB technology - Webthesis, Zugriff am November 4, 2025, https://webthesis.biblio.polito.it/21180/1/tesi.pdf
58. Visual Servoing Approach to Autonomous UAV Landing on a Moving Vehicle - MDPI, Zugriff am November 4, 2025, https://www.mdpi.com/1424-8220/22/17/6549
59. High-Precision Landing on a Moving Platform Based on Drone Vision Using YOLO Algorithm - MDPI, Zugriff am November 4, 2025, https://www.mdpi.com/2504-446X/9/4/261
60. Precise Interception Flight Targets by Image-based Visual Servoing of Multicopter - arXiv, Zugriff am November 4, 2025, https://arxiv.org/html/2409.17497v1
61. Towards Safe Mid-Air Drone Interception: Strategies for Tracking & Capture - arXiv, Zugriff am November 4, 2025, https://arxiv.org/html/2405.13542v2
62. Precise Interception Flight Targets by Image-based Visual Servoing of Multicopter - arXiv, Zugriff am November 4, 2025, https://arxiv.org/html/2409.17497v2
63. Exploring the 2023 U.S. Directive on Autonomy in Weapon Systems - CEBRI, Zugriff am November 4, 2025, https://cebri.org/revista/en/artigo/114/exploring-the-2023-us-directive-on-autonomy-in-weapon-systems
64. The Impact of Digital Technology on International Humanitarian Law: Ethical and

Legal Implications of Autonomous Weapons Systems | European Journal of Law and Political Science, Zugriff am November 4, 2025, https://www.ej-politics.org/index.php/politics/article/view/182/221
65. Full article: Supporting Ethical Decision-Making for Lethal Autonomous Weapons, Zugriff am November 4, 2025, https://www.tandfonline.com/doi/full/10.1080/15027570.2024.2366094
66. The Kargu-2 Autonomous Attack Drone: Legal & Ethical Dimensions ..., Zugriff am November 4, 2025, https://lieber.westpoint.edu/kargu-2-autonomous-attack-drone-legal-ethical/
67. Potential Use of Autonomous Lethal Drone in Libya - Defence.AI, Zugriff am November 4, 2025, https://defence.ai/industry-news/ai-lethal-drone-libya/
68. Libya, The Use of Lethal Autonomous Weapon Systems - How does law protect in war?, Zugriff am November 4, 2025, https://casebook.icrc.org/print/21244
69. Staying in the loop: Human supervisory control of weapons - ResearchGate, Zugriff am November 4, 2025, https://www.researchgate.net/publication/325793611_Staying_in_the_loop_Human_supervisory_control_of_weapons
70. (PDF) A survey of human-in-the-loop for machine learning - ResearchGate, Zugriff am November 4, 2025, https://www.researchgate.net/publication/360707820_A_survey_of_human-in-the-loop_for_machine_learning
71. Autonomous weapons and stability Scharre, Paul - This electronic ..., Zugriff am November 4, 2025, https://kclpure.kcl.ac.uk/portal/files/129451536/2020_Scharre_Paul_1575997_ethesis.pdf
72. Providing Autonomous Flight to drone - ROS 1 / ROS 2 - PX4 Discussion Forum, Zugriff am November 4, 2025, https://discuss.px4.io/t/providing-autonomous-flight-to-drone/45310
73. ROS/Gazebo based simulation of co-operative UAVs? - ARPI - UNIPI, Zugriff am November 4, 2025, https://arpi.unipi.it/retrieve/e0d6c930-b66f-fcf8-e053-d805fe0aa794/Mesas2018postprint.pdf
74. COMPARATIVE REVIEW OF DRONE SIMULATORS, Zugriff am November 4, 2025, https://ela.kpi.ua/bitstreams/6ac76caf-05ce-4dc0-95cf-c2b0644958b1/download
75. Basic OpenCV with ROS for Drones - YouTube, Zugriff am November 4, 2025, https://www.youtube.com/watch?v=WtJtkg6NLmQ
76. Lethal Autonomous Weapons Systems & International Law: Growing Momentum Towards a New International Treaty | ASIL, Zugriff am November 4, 2025, https://www.asil.org/insights/volume/29/issue/1
77. Drone Control Using Pixhawk and ROS | Office of Undergraduate Research, Zugriff am November 4, 2025, https://undergraduateresearch.wvu.edu/symposia/summer-2021/past-symposia/-poster-presentations/engineering-poster-presentations-51-70/drone-control-using-pixhawk-and-ros
78. ROS 2 | PX4 Guide (main), Zugriff am November 4, 2025,

https://docs.px4.io/main/en/ros2/
79. A Step-by-Step Guide to Creating a Robust Autonomous Drone Testing Pipeline - arXiv, Zugriff am November 4, 2025, https://arxiv.org/html/2506.11400v1
80. DIY Autonomous Drone with Raspberry Pi, Pixhawk, and Machine Learning : r/diydrones, Zugriff am November 4, 2025, https://www.reddit.com/r/diydrones/comments/1h93h0i/diy_autonomous_drone_with_raspberry_pi_pixhawk/
81. NVIDIA® Jetson Comparison: Nano, TX2 NX, Xavier NX, AGX, Orin - Latest News from Seeed Studio, Zugriff am November 4, 2025, https://www.seeedstudio.com/blog/nvidia-jetson-comparison-nano-tx2-nx-xavier-nx-agx-orin/
82. Orin Nano 8GB or Xavier NX 16GB for a Fighter UAV: ? : r/JetsonNano - Reddit, Zugriff am November 4, 2025, https://www.reddit.com/r/JetsonNano/comments/1it4pft/orin_nano_8gb_or_xavier_nx_16gb_for_a_fighter_uav/
83. Aerostack2: A Software Framework for Developing Multi-robot Aerial Systems - arXiv, Zugriff am November 4, 2025, https://arxiv.org/html/2303.18237v2
84. Autonomous Drones with ROS and simulation in unity. - GitHub, Zugriff am November 4, 2025, https://github.com/AbhinavUtkarsh/Autonomous-Drones-with-ROS
85. MikeS96/autonomous_landing_uav: ROS packages of the Autonomous landing system of a UAV in Gazebo - GitHub, Zugriff am November 4, 2025, https://github.com/MikeS96/autonomous_landing_uav
86. ctu-mrs/mrs_uav_system: The entry point to the MRS UAV system. - GitHub, Zugriff am November 4, 2025, https://github.com/ctu-mrs/mrs_uav_system
87. ROS (1) with MAVROS Installation Guide | PX4 Guide (main), Zugriff am November 4, 2025, https://docs.px4.io/main/en/ros/mavros_installation
88. Comparison between Gazebo and AirSim for HITL - PX4 Discussion Forum, Zugriff am November 4, 2025, https://discuss.px4.io/t/comparison-between-gazebo-and-airsim-for-hitl/7304
89. Comparative Study of Drone Simulation Frameworks: ArduPilot with Gazebo vs. AirSim with PX4 - OPUS Open Portal to University Scholarship, Zugriff am November 4, 2025, https://opus.govst.edu/cgi/viewcontent.cgi?article=1646&context=research_day
90. Choose a Simulator - Robotics Knowledgebase, Zugriff am November 4, 2025, https://roboticsknowledgebase.com/wiki/robotics-project-guide/choose-a-sim/
91. Introduction to Gazebo | Advanced Drone Simulation - YouTube, Zugriff am November 4, 2025, https://www.youtube.com/watch?v=-ttibzzc0JQ
92. Simulate a Camera on a Drone with Gazebo - YouTube, Zugriff am November 4, 2025, https://www.youtube.com/watch?v=ZfiJhiA1ksA
93. pixhawk-flight-controller · GitHub Topics, Zugriff am November 4, 2025, https://github.com/topics/pixhawk-flight-controller?o=asc&s=updated
94. Zhefan-Xu/CERLAB-UAV-Autonomy: [CMU] A Versatile and ... - GitHub, Zugriff am November 4, 2025, https://github.com/Zhefan-Xu/CERLAB-UAV-Autonomy