

SafeLand: Safe Autonomous Landing in Unknown Environments with Bayesian Semantic Mapping

Markus Gross^{1,2,3,*}, Andreas Greiner¹, Sai B. Matha¹,
Felix Soest¹, Daniel Cremers^{2,3}, Henri Meek¹

¹Fraunhofer Institute IVI ²TU Munich ³Munich Center for Machine Learning

Abstract. Autonomous landing of uncrewed aerial vehicles (UAVs) in unknown, dynamic environments poses significant safety challenges, particularly near people and infrastructure, as UAVs transition to routine urban and rural operations. Existing methods often rely on prior maps, heavy sensors like LiDAR, static markers, or fail to handle non-cooperative dynamic obstacles like humans, limiting generalization and real-time performance. To address these challenges, we introduce SafeLand, a lean, vision-based system for safe autonomous landing (SAL) that requires no prior information and operates only with a camera and a lightweight height sensor. Our approach constructs an online semantic ground map via deep learning-based semantic segmentation, optimized for embedded deployment and trained on a consolidation of seven curated public aerial datasets (achieving 70.22% mIoU across 20 categories), which is further refined through Bayesian probabilistic filtering with temporal semantic decay to robustly identify metric-scale landing spots. A behavior tree then governs adaptive landing, iteratively validates the spot, and reacts in real time to dynamic obstacles by pausing, climbing, or rerouting to alternative spots, maximizing human safety. We extensively evaluate our method in 200 simulations and 60 end-to-end field tests across industrial, urban, and rural environments at altitudes up to 100m, demonstrating zero false negatives for human detection. Compared to the state of the art, SafeLand achieves sub-second response latency, substantially lower than previous methods, while maintaining a superior success rate of 95%. To facilitate further research in aerial robotics, we release SafeLand’s segmentation model as a plug-and-play ROS package, available at <https://github.com/markus-42/SafeLand>.

1 Introduction

Autonomous landing of uncrewed aerial vehicles (UAVs) is a safety-critical task, especially in proximity to people and infrastructure [14, 53]. As UAVs move from isolated use cases toward dense deployments in logistics, inspection, and emergency response [49], regulatory frameworks such as the Specific Operations Risk Assessment (SORA) [14] impose stringent risk constraints. Landing systems must therefore ensure safety under diverse and only partially known conditions while

*Corresponding author: markus.gross@tum.de.

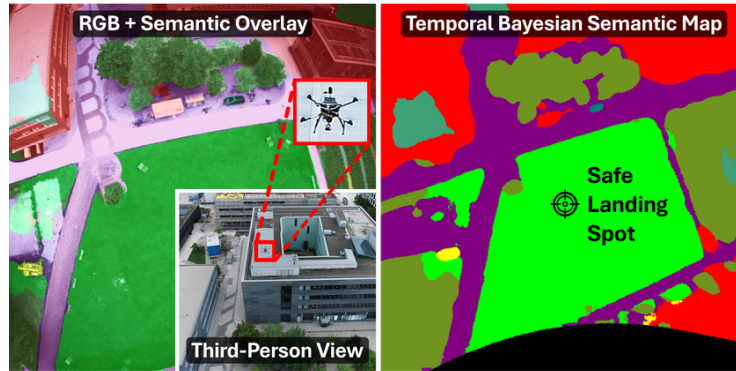


Fig. 1: SafeLand predicts a semantic map, projects it to a metric ground plane, filters it probabilistically over time to select a landing spot, and uses a behavior tree for safe real-time landing in unknown and dynamic environments.

remaining lightweight and cost-efficient. Software and vision-based methods are central to this development, as they offer precise relative positioning from compact, low-power sensors, and are applicable across a multitude of UAV platforms [14, 49].

To this end, most vision-centric methods for *safe autonomous landing (SAL)* are marker-based [49]: they achieve high accuracy and robustness on prepared landing pads by exploiting carefully designed markers and controlled infrastructure [59]. However, these systems assume instrumented, static landing sites and thus do not generalize to generic outdoor operations, such as emergency landings in populated areas, or slowly changing environments, such as agricultural fields. Moreover, beyond marker-based methods, existing works employ markerless, hybrid, and multi-sensor approaches, which exhibit systematic limitations [14]: many methods (i) assume prior maps, (ii) lack explicit metric reasoning due to the scale-ambiguous nature of camera-based projective geometry, (iii) depend on heavy and power-consuming depth sensors such as LiDAR or radar, (iv) treat humans as generic obstacles if at all, and (v) are rarely validated end-to-end in sophisticated field experiments.

Motivated by these gaps, we introduce *SafeLand* (Fig. 1), a lightweight, vision-based SAL system that builds an online semantic ground map from deep learning-based semantic segmentation and height-above-ground-level sensing, refines it via Bayesian temporal-probabilistic filtering, and uses a behavior tree to adapt landing decisions to non-cooperative dynamic obstacles, including people. Since generic vision foundation models underperform on aerial semantic segmentation [43], we train a network on curated public aerial datasets and release it as a plug-and-play ROS package for the community.

In summary, our contributions are as follows:

- We present SafeLand, a lightweight, vision-based UAV landing solution that operates with a minimal sensor setup, requires no prior information about the landing site, and ensures safety for humans and infrastructure.
- We train a robust aerial semantic segmentation model from 7 curated datasets, achieving 70.22 mIoU, which we further optimize for embedded deployment using the TensorRT engine.
- We evaluate SafeLand in 200 simulations and 60 diverse field tests across industrial, urban, and rural environments at 25 m to 100 m altitudes, with deliberate interventions by non-cooperative dynamic obstacles, such as humans.
- SafeLand substantially outperforms baselines with sub-second response latency and a 95 % landing success rate.
- Upon publication, we will release SafeLand’s segmentation model as a plug-and-play ROS package to facilitate further research in vision-based aerial robotics.

2 Related Work

Apart from methods that rely on depth sensors, such as LiDAR and radar [9, 29, 33, 39, 48, 56, 57], which often add significant weight and power consumption, vision-based methods have gained prominence as lightweight, cost-effective alternatives for SAL in unknown environments. These vision-centric approaches process camera imagery to detect suitable landing sites, and are often categorized as traditional image processing for feature extraction, such as edge detection or texture analysis [16, 24, 55], supervised zone-classification based on visual cues [3, 5, 22, 26, 37], or more advanced deep learning-driven semantic segmentation to delineate terrain types [4, 6, 17, 44, 52]. A prevalent limitation across many is the dependence on priors, such as pre-existing maps of potential landing areas [8, 27, 38, 45] or structured environments with instrumented landing pads [39, 56, 57], which restricts their applicability to truly dynamic or unprepared scenarios where environmental changes or lack of infrastructure demand greater flexibility [14]. Furthermore, to address the inherent scale ambiguity in camera-based projective geometry, several methods integrate metric compensation strategies, including assumptions of flat ground for homography-based transformations [30, 60], depth cameras [60], or fusion with complementary data sources like inertial measurements and altitude sensors to recover absolute scales [3, 6, 16, 18, 24, 35, 44, 54, 55], enabling precise sizing of landing spots relative to UAV dimensions.

Importantly, safety considerations, particularly for people as non-cooperative dynamic obstacles, are underexplored in the literature and, when addressed, are typically handled only in a subset of works through techniques like population density estimation from aerial views [8, 19] or real-time visual tracking to monitor and predict human movements [12, 18, 30, 35, 45, 54]. Notably, comprehensive validation remains a gap [14]: to the best of our knowledge, only Safe2Ditch [30] conducts so-

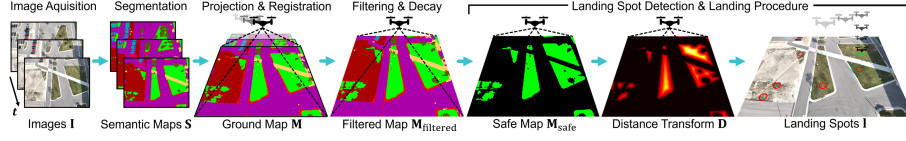


Fig. 2: Chronological processing steps of SafeLand. An overview is provided in Sec. 3.1.

phisticated end-to-end field tests under real-world conditions, demonstrating practical functionality with human interference, whereas the majority evaluate individual components, such as standalone perception modules, or confine assessments entirely to simulations [3–6, 8, 9, 14, 16–19, 22, 24, 26, 27, 29, 33, 35, 37–39, 44, 45, 54–57], limiting insights to isolated challenges, such as lighting conditions, without considering holistic system viability.

In summary, SafeLand follows this line of vision-based SAL but differs in three key aspects. It trains a dedicated aerial segmentation model from complementary datasets to handle diverse unknown environments, restores metric scale using only a lightweight height above ground level (AGL) sensor instead of heavy depth hardware or prior maps, and explicitly targets human safety with conservative Bayesian filtering of person detections, validated in 200 simulations and 60 field tests across industrial, urban, and rural sites up to 100 m AGL.

3 Methodology

3.1 Method Overview

SafeLand (see Fig. 2) uses only a nadir-pointing camera and an AGL sensor. It applies deep learning-based semantic segmentation on temporal camera frames, projects the resulting semantic probabilities onto a discrete, metrically scaled ground plane to obtain a semantic ground map, and iteratively filters this map via Bayesian inference and semantic decay for robust perception. We then extract safe landing spot candidates, and a behavior tree-based landing procedure navigates the UAV toward the selected spot while re-selecting a new spot if the current one becomes invalid. This pipeline continuously evaluates the environment, reacts in real time to non-cooperative obstacles, and operates without prior map information or heavy sensors, supporting generic deployment.

3.2 Image Acquisition and Semantic Segmentation

A nadir-pointing RGB camera captures real-time imagery for semantic segmentation. Robust perception in previously unseen and diverse terrain requires diverse training data, yet existing aerial datasets are individually too narrow, and current vision foundation models for semantic segmentation underperform in the aerial domain [43]. We therefore consolidate seven open-source

datasets [7, 10, 25, 31, 40, 47, 51] that provide aerial imagery with semantic segmentation ground-truth and curate their labels into 20 unified classes.

We train a SegFormer MiT-B3 model [58] with 47.3 million parameters and obtain 70.22 % mIoU, detailed in Tab. 1. We select SegFormer for its balance between accuracy and efficiency, which enables deployment on resource-constrained onboard hardware. This model is further optimized for embedded deployment using the TensorRT inference engine [42] with FP16 precision calibration, increasing standalone throughput to 12 FPS.

Formally, let $\mathcal{C}$ be the set of semantic classes summarized in Tab. 1. The camera image $\mathbf{I} \in \mathbb{R}^{M \times N \times 3}$ is segmented as

$$\mathbf{S} = \Phi_s(\mathbf{I}) , \quad (1)$$

where Φ_s denotes the segmentation model, $\mathbf{S} \in \mathbb{R}^{M \times N \times C}$ is the predicted map, $M \times N$ represents the image resolution, and $C = |\mathcal{C}|$ is the number of semantic classes.

Overall, semantic segmentation relies only on a lightweight and cost-efficient camera, removes the need for prior map information, facilitates generalization to diverse environments, and improves safety by explicitly modeling humans.

3.3 Projection and Registration

These steps follow the method described in [23], and assume a flat ground and AGL measurements. In summary, the undistorted semantic map is projected onto a ground plane using the UAV’s pose and height, in addition to camera intrinsics and extrinsics, yielding a discrete semantic ground map $\mathbf{M} \in \mathbb{R}^{X \times Y \times C}$ with fixed size and metric scale.

Formally, a homogeneous pixel coordinate $\mathbf{s} \in \mathbb{P}^2$ on the predicted semantic map $\mathbf{S}$ and its corresponding homogeneous 3D point $\tilde{\mathbf{m}} \in \mathbb{P}^3$ on the semantic ground map $\mathbf{M}$ are related by

$$\mathbf{s} = [K \mid \mathbf{0}] \cdot \mathbf{T}_{\text{cam} \leftarrow \text{world}} \cdot \tilde{\mathbf{m}} , \quad (2)$$

where $\mathbf{0} \in \mathbb{R}^3$ is the null vector, $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ are camera intrinsics, and $\mathbf{T}_{\text{cam} \leftarrow \text{world}} \in \text{SE}(3)$ denotes a rigid transformation from planar world coordinates to camera coordinates [23].

Subsequent registration aligns earlier maps with the current one, by transforming a Cartesian point $\mathbf{m}_{t-1} \in \mathbb{R}^3$ via

$$\mathbf{M}_t(\mathbf{m}_{t-1}) = \mathbf{M}_{t-1}(\mathbf{H}_{t \leftarrow (t-1)} \cdot \mathbf{m}_{t-1}) , \quad (3)$$

where $\mathbf{M}_t$ and $\mathbf{M}_{t-1}$ denote the ground maps at times t and $(t-1)$, respectively. $\mathbf{H}_{t \leftarrow (t-1)} \in \mathbb{R}^{3 \times 3}$ is a homography that warps points from the previous map’s coordinate system to the current one, which is computed from the camera extrinsics at times t and $(t-1)$. In practice, four or more corresponding points are sufficient to estimate this homography whenever the extrinsics change. For more details, see [23].

3.4 Bayesian Filtering and Semantic Decay

The segmentation model’s predictions remain subject to residual errors and dataset-induced biases. We therefore iteratively refine the semantic ground map $\mathbf{M}$ over time, enhancing robustness and safety of subsequent landing spot detection.

Formally, we apply Bayesian inference and treat the model’s softmax outputs as unnormalized conditional class probabilities that serve as proxies for likelihood terms in the Bayesian update. This filtering process updates the belief about the state of each pixel in the semantic ground map based on new observations from the segmentation model Φ_s . Let $P(c \mid \mathbf{M}_t)$ be the posterior probability of class c given the segmentation map $\mathbf{M}_t$ at time t . The filtering process is expressed as

$$P(c \mid \mathbf{M}_t) = \frac{P(\mathbf{M}_t \mid c) \cdot P(c \mid \mathbf{M}_{t-1})}{P(\mathbf{M}_t)}, \quad (4)$$

where $P(\mathbf{M}_t \mid c)$ represents the likelihood of observing the semantic ground map $\mathbf{M}_t$ given class c , $P(c \mid \mathbf{M}_{t-1})$ is the prior probability of class c at time $t - 1$, and $P(\mathbf{M}_t)$ is the evidence (marginal likelihood). While the evidence is often omitted when only relative posterior scores are needed, we compute it explicitly by normalizing the posterior to ensure numerical stability. We initialize with a uniform prior.

Subsequently, semantic decay incrementally updates the posterior beliefs over time, incorporating new posteriors by applying a decay factor to downweight the contribution of past observations. Let α be the decay factor, and $\mathbf{M}_t$ be the refined semantic ground map at time t . This update is formulated as

$$\mathbf{M}_t(\mathbf{m}, c) = \alpha \cdot \mathbf{M}_{t-1}(\mathbf{m}, c) + (1 - \alpha) \cdot P(c \mid \mathbf{M}_t(\mathbf{m}, c)), \quad (5)$$

where $\mathbf{M}_t(\mathbf{m}, c)$ and $\mathbf{M}_{t-1}(\mathbf{m}, c)$ are the probabilities of class c at position $\mathbf{m}$ in the refined map at time t and $(t - 1)$, respectively, and α is the decay factor, which is empirically set to $\alpha = 0.1$. Eqn. 5 also implies that, if no updates are received for α^{-1} iterations, the semantic estimate decays to 0, ensuring that stale evidence is forgotten and the map remains up to date with recent observations.

Furthermore, to ensure safety in our proposed method, we treat human detections conservatively by setting the class probability of any predicted person to $P(\mathbf{M}_t \mid c) = 1$, where $c \hat{=} \textit{person}$. Moreover, semantic decay for persons is only applied if a different semantic class is predicted at the same position, ensuring that their probability does not decay to 0 when no updates are received.

Formally, the filtered map $\mathbf{M}_{\text{filtered}} \in \mathbb{R}^{X \times Y \times 1}$ is determined by selecting the class with the highest probability at each location $\mathbf{m}$ as the definitive class for that location:

$$\mathbf{M}_{\text{filtered}}(\mathbf{m}) = \arg \max_c \mathbf{M}_t(\mathbf{m}, c). \quad (6)$$

By applying Bayesian filtering with semantic decay, we mitigate segmentation errors, ensure robust performance in unknown environments, and enhance human safety.

Table 1: Intersection over union (IoU) and mean IoU (mIoU) in [%] for semantic segmentation (Sec. 3.2). To ensure fair comparison, we retain the train/val/test splits of the original datasets [7, 10, 25, 31, 40, 47, 51]. Thus, some classes appear only in the training split and are absent from val/test, so their performance cannot be quantified and is marked as n/a. However, for these classes, we (1) qualitatively assess them and (2) confirm their high practical performance by successfully landing on them in real-world field tests (Sec. 4.2). The lower Bicycle score is consistent with its sparse occurrence in the data.

mIoU	Road	Dirt	Gravel	Rock	Grass	Vegetation	Tree	Obstacle	Animal	Person	Bicycle	Vehicle	Water	Boat	Wall	Roof	Sky	Drone	Train-Track	Background	
70.22	92.85	n/a	n/a	n/a	n/a	89.18	96.41	81.52	39.14	81.74	70.09	1.88	69.18	83.30	n/a	83.49	81.05	97.40	35.49	50.61	—

3.5 Landing Spot Detection

A safe landing spot is identified in three steps (see Fig. 2). First, we define and rank semantic classes suitable for safe landing (e.g., grass, dirt, gravel), denoted as $\mathcal{C}_{\text{safe}}$, and only consider these classes in the safe semantic ground map $\mathbf{M}_{\text{safe}}$:

$$\mathbf{M}_{\text{safe}} = \{\mathbf{M}_{\text{filtered}}(c) \mid c \in \mathcal{C}_{\text{safe}}\} . \quad (7)$$

We then apply a distance transform [13] to each segment individually. Illustrated in Fig. 2, each pixel represents the least distance to its segment border, expressed as $\mathbf{D} \in \mathbb{R}^{X \times Y \times 1}$:

$$\mathbf{D} = \text{distance_transform}(\mathbf{M}_{\text{safe}}) . \quad (8)$$

Since the distances are measured on a metric scale (see Sec. 3.3), we define an adjustable safety radius r_{safe} that accounts for the physical UAV dimensions. The segment with the largest distance value that exceeds this radius is chosen as the landing segment. The coordinates of this maximum value define the center of the landing spot, denoted by $\mathbf{l} \in \mathbb{Z}^2$:

$$\mathbf{l} = \underset{\mathbf{D}(x,y) \geq r_{\text{safe}}}{\arg \max} \mathbf{D}(x,y) . \quad (9)$$

All segments that satisfy these constraints are stored as a history of safe landing spots. The ranking of semantic classes and the choice of the point of maximum distance within a safe segment are conservative design parameters that can be replaced by more adaptive criteria if required. By utilizing a metric distance transform, we naturally account for UAV size constraints ($r_{\text{safe}} = 3\text{ m}$ in experiments), enabling safe landings without prior map information, such as designed markers.

3.6 Landing Procedure

Once a safe landing spot $\mathbf{l}$ (Eqn. 9) has been identified, the landing procedure is governed by a behavior tree [11], as illustrated in Fig. 3. Behavior trees provide a modular and reactive decision-making framework, enabling continuous

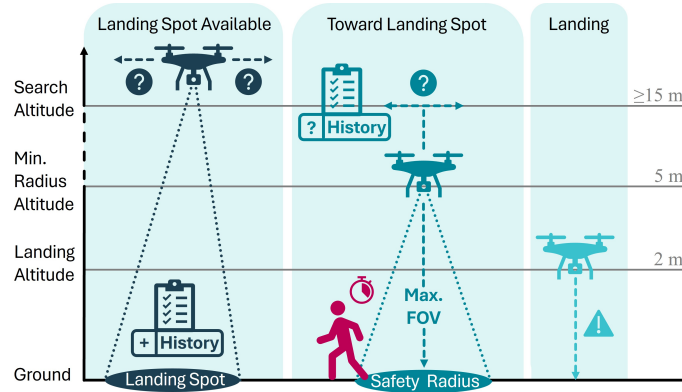


Fig. 3: Landing procedure. A behavior tree (Sec. 3.6) executes three distinct sequences that ensure safe, real-time landing in unknown and dynamic environments.

reassessment of the environment while maintaining deterministic execution. This is particularly advantageous for safe autonomous landing, where environmental conditions may change throughout the descent and immediate reactions to dynamic obstacles are required. The proposed behavior tree consists of three main execution sequences:

Landing Spot Available Sequence. The first sequence continuously searches for valid landing locations while the UAV remains at or above an adjustable search altitude of 15 m (Fig. 3). During this phase, the complete perception pipeline, from semantic segmentation to landing spot detection (Sections 3.2–3.5), is executed repeatedly to maintain an up-to-date representation of the environment. Whenever one or more valid landing spots are detected, their locations are stored in a history of validated landing sites. This history enables the system to quickly revisit alternative candidates should the currently selected landing location become invalid later. If no suitable landing site is available, the UAV continues along its current heading as a simple exploration strategy. Since the search behavior is independent of the remaining landing pipeline, this can readily be replaced by more sophisticated mission-specific exploration or path-planning algorithms.

Toward Landing Spot Sequence. Once a landing location has been selected, the UAV first performs level flight toward a waypoint directly above the landing spot before initiating its descent toward the predefined landing altitude (Fig. 3). Importantly, selecting a landing site does not constitute a final commitment. Instead, the system continuously re-evaluates the safety of the selected location throughout the descent using the perception pipeline. If the landing site becomes invalid, e.g., due to newly observed terrain or the appearance of dynamic obstacles, the descent is immediately aborted, and the UAV climbs back to the search

altitude. From there, the system either identifies a new landing location or retrieves a previously validated candidate from the landing history.

This continuous reassessment is performed until the UAV reaches the minimum radius altitude of 5 m. At this point, the projected safety radius fully covers the camera’s field of view (Fig. 3), eliminating the need to repeatedly evaluate the surrounding terrain. Consequently, the system focuses exclusively on monitoring dynamic obstacles, as they represent the only remaining source of uncertainty. Whenever a person or other non-cooperative obstacle enters the landing area, the UAV pauses its descent for up to 5 seconds. If the landing zone clears during this interval, the descent resumes immediately. Otherwise, the landing attempt is aborted, and the UAV returns to the search altitude, where a new landing decision is initiated.

Landing Sequence. The final sequence is entered once the UAV reaches the landing altitude of 2 m. At this stage, the UAV commits to the landing and no longer performs additional decision-making. This design choice intentionally balances safety with flight stability. At very low altitudes, aerodynamic ground effects become increasingly pronounced, while the semantic segmentation model exhibits reduced reliability below approximately 2 m owing to the limited availability of low-altitude training imagery. Moreover, by the time this altitude is reached, the selected landing site has already remained valid throughout the entire descent, substantially increasing confidence that the landing can be completed safely. Consequently, introducing further decision points at this stage would provide limited benefit while unnecessarily increasing system complexity during the most critical phase of the maneuver.

Overall, the proposed behavior tree enables reactive, real-time landing decisions while remaining independent of prior maps or pre-defined landing infrastructure. By continuously validating candidate landing sites and explicitly accounting for dynamic human activity until shortly before touchdown, the framework provides a practical and safety-oriented control strategy for autonomous landing in previously unseen environments.

4 Experimental Results

4.1 Simulation Experiments

Setup: We adopt a software-in-the-loop (SIL) setup using the OCTAS v1.0 simulation framework [50], integrated with ROS2 Humble [32] and PX4 v1.15.4 [34], thereby closely mirroring the software stack employed during the real-world field experiments. This enables us to evaluate the complete perception and landing pipeline under controlled yet realistic operating conditions while ensuring a seamless transition between simulation and hardware deployment.

Throughout the experiments, we define “grass” as the safe semantic class $\mathcal{C}_{\text{safe}}$ for landing. This choice reflects the simulated environment but can readily be

adapted to different semantic categories depending on mission-specific operational requirements. Since the semantic segmentation model (Sec. 3.2) is trained exclusively on real-world imagery and is therefore not optimized for synthetic data, we use the simulator’s semantic ground-truth annotations to isolate the evaluation of the proposed landing framework from the domain gap between simulated and real imagery.

To evaluate the robustness of the proposed approach under dynamic conditions, we perform 200 independent landing trials from an initial altitude of 50 m. During each trial, up to 10 non-cooperative dynamic obstacles, such as humans, are introduced to assess the system’s ability to respond to changing environments by waiting or rerouting while maintaining safe landing execution.

Results: The first column of Fig. 6 illustrates a representative simulation sequence. The proposed Bayesian temporal-probabilistic filtering progressively refines the semantic scene representation, enabling the selection of a valid landing site within the predefined safe semantic classes despite continuously changing observations. As the UAV descends, the selected landing location remains consistent while still adapting to newly observed information.

Whenever a non-cooperative obstacle entered the predefined safety radius, the behavior tree correctly suspended the landing procedure until the area became safe again or, if necessary, selected an alternative landing location. Across all 200 trials, every landing was completed successfully without a single failed landing or incorrect rerouting decision. These results demonstrate that the proposed framework reliably balances stable landing-site selection with rapid reactions to dynamic obstacles, providing a robust foundation before transferring the system to real-world deployments.

4.2 Field Experiments

Setup: To validate the proposed framework under real operating conditions, we implement the complete system in ROS2 Humble [32] using the hardware configuration shown in Fig. 4. The onboard Jetson companion computer receives RGB images from the camera via Ethernet and communicates with the Pixhawk flight controller through a UART uXRCE-DDS Bridge [1]. The Pixhawk further interfaces with the GNSS receiver via UART and the above-ground-level (AGL) sensor through I2C. Throughout all experiments, the UAV pose is obtained from the Pixhawk Extended-Kalman-Filter (EKF) estimator, which fuses measurements from the IMU, GNSS receiver, and AGL sensor.

The experimental campaign comprises a total of 60 autonomous landing flights conducted across three representative operational environments, including industrial, urban, and rural scenes, with 20 flights performed in each environment. To evaluate robustness across different mission profiles, flights are conducted from altitudes of 25 m, 50 m, and 100 m. We define “grass” and “gravel” as safe semantic categories $\mathcal{C}_{\text{safe}}$, although these semantic definitions can be readily adapted to mission-specific requirements. To evaluate the safety mechanisms of the proposed



Fig. 4: Hardware configuration. **Left:** Nadir-pointing RGB camera FLIR Blackfly S-PGE-31S4C-C [15] and the height AGL sensor Lightware SF11/C [28]. **Center:** Quadcopter PM Q-685 [36] custom frame, equipped with NVIDIA Jetson AGX Orin [41], GNSS module SimpleRTK2B Pro [2], and flight controller Pixhawk Cube Orange with its IMUs [46]. **Right:** Block diagram of the SafeLand hardware setup.

framework, up to three participants actively enter the designated landing zone during the landing procedure, forcing the system to react to dynamic human presence.

Results: Representative examples of the field experiments are shown in Fig. 6. Out of the 60 autonomous landing attempts, 57 were completed successfully, corresponding to a landing success rate of 95%. The remaining three flights were safely aborted due to transient communication failures between the Pixhawk flight controller and the Jetson companion computer, which are external to the proposed perception and decision-making framework.

The experiments further demonstrate reliable operation in the presence of dynamic obstacles. Whenever individuals entered the landing zone, the behavior tree either paused the descent until the area became safe again or rerouted the UAV to an alternative landing location when the originally selected site remained occupied. During two flights, the semantic segmentation model temporarily produced false-positive human detections within the landing zone, resulting in conservative delays in the landing procedure. These false detections were subsequently corrected through Bayesian temporal filtering and semantic decay as additional observations became available, thereby allowing both landings to be completed successfully. Importantly, no false negatives occurred throughout the entire experimental campaign, meaning that every person entering the landing zone was successfully detected. This conservative behavior highlights the safety-oriented design of the proposed framework, in which temporary delays are preferred over potentially unsafe landing decisions.

Across all evaluated environments and flight altitudes, the semantic segmentation model consistently identified suitable landing regions under varying scene layouts and illumination conditions. Combined with Bayesian filtering and semantic decay, the resulting semantic world representation remained both stable and responsive, enabling the behavior tree to make reliable landing decisions while

Table 2: Baseline comparison of autonomous landing methods against requirements identified in recent survey literature [14, 49] (see Sec. 4.3). – marks ambiguous references.

Reference	No Prior	Metric	No Lid./Rad.	People	Field
[39, 56, 57]	✗	✓	✗	✗	✗
[27]	✗	✗	✓	✗	✗
[38]	✗	✓	–	✗	✗
[6, 44]	✗	✓	✓	✗	✗
[4, 9, 17]	✓	–	✓	✗	✗
[37]	✓	✗	✓	✗	✗
[5, 22, 26]	✓	✗	✓	✗	✗
[29, 33]	✓	✓	✗	✗	✗
[3]	✓	✓	–	✗	✗
[16, 24, 52, 55, 60]	✓	✓	✓	✗	✗
[8, 45]	✗	✓	✓	✓	✗
[48]	✓	✓	✗	✗	✓
[19]	✓	✗	✓	✓	✗
[12, 18, 35, 54]	✓	✓	✓	✓	✗
Safe2Ditch [30]	✗	✓	✓	✓	✓
SafeLand (ours)	✓	✓	✓	✓	✓

continuously accounting for dynamic human activity. Running on the onboard Jetson, the complete perception and decision-making pipeline achieves end-to-end processing rates of 1.3 Hz (30 W), 2.0 Hz (50 W), and 3.2 Hz (MAXN mode), demonstrating that the proposed system satisfies the computational requirements for real-time onboard deployment.

4.3 Baseline Comparison

Existing work on safe autonomous landing (Sec. 2) together with recent comprehensive surveys [14, 49] identifies several fundamental requirements for practical vision-based landing systems. Such systems should (1) operate without prior information about candidate landing zones to enable deployment in unknown and dynamic environments, (2) estimate landing areas in metric units to account for different UAV dimensions, (3) rely on lightweight sensing modalities such as RGB cameras rather than heavier and more power-intensive sensors including LiDAR or radar, (4) explicitly prioritize human safety throughout the landing procedure, and (5) demonstrate complete end-to-end validation through real-world flight experiments. Table 2 summarizes these requirements and shows that, to the best of our knowledge, our method is the first to satisfy all of them simultaneously.

Among existing approaches, Safe2Ditch [30] is the only work that presents a complete autonomous landing system together with comprehensive field validation. We therefore consider it the strongest available baseline, despite relying on a database of pre-selected landing sites as prior information. This assumption fundamentally differs from our problem formulation, which deliberately avoids prior environmental knowledge to enable deployment in previously unseen environments.

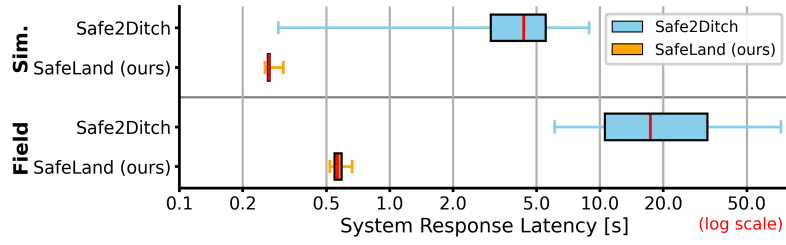


Fig. 5: System response latencies (see Sec. 4.3) on dynamic obstacles for Safe2Ditch [30] and SafeLand (ours). We substantially outperform Safe2Ditch by providing sub-second response latency as a critical capability for unknown and dynamic environments.

To evaluate responsiveness in dynamic environments, we compare the response latency, defined as the time between a non-cooperative obstacle entering the currently selected landing zone and the triggering of a safety intervention. This metric directly reflects the ability of a landing system to react to unforeseen hazards during the critical final landing phase, where delayed decisions reduce the available altitude and maneuvering time. Since Safe2Ditch reports response latencies only graphically, we digitize the published plots to recover the reported measurements. To ensure a fair comparison, we apply an identical aggregation procedure to both methods by averaging simulation results across scenarios containing one to ten non-cooperative obstacles and averaging field measurements across all reported flight altitudes.

Figure 5 shows that our method consistently achieves substantially lower response latency, providing sub-second reactions to dynamic obstacles. This improvement follows directly from the architectural differences between the two systems. Safe2Ditch evaluates dynamic obstacles with respect to a database of pre-selected landing sites, requiring tracked objects to be geolocated before rerouting decisions can be made. Safe2Ditch identifies this target geolocation stage as the dominant contributor to the overall response latency due to localization uncertainty. In contrast, our method performs safety assessment directly on the current semantic scene representation and therefore does not require prior landing-site databases or an intermediate target geolocation stage. Besides enabling operation in unknown environments, this architectural simplification substantially reduces the system response time.

Finally, we compare failure rates. In simulation, both methods achieve effectively zero failures. During field experiments, Safe2Ditch reports one failure in sixteen flights caused by obstacle misclassification, whereas our system experienced three failures in sixty flights, corresponding to a success rate of 95%. Importantly, all observed failures were traced to transient communication issues between the Pixhawk flight controller and the Jetson companion computer and are therefore independent of the proposed perception and landing framework.

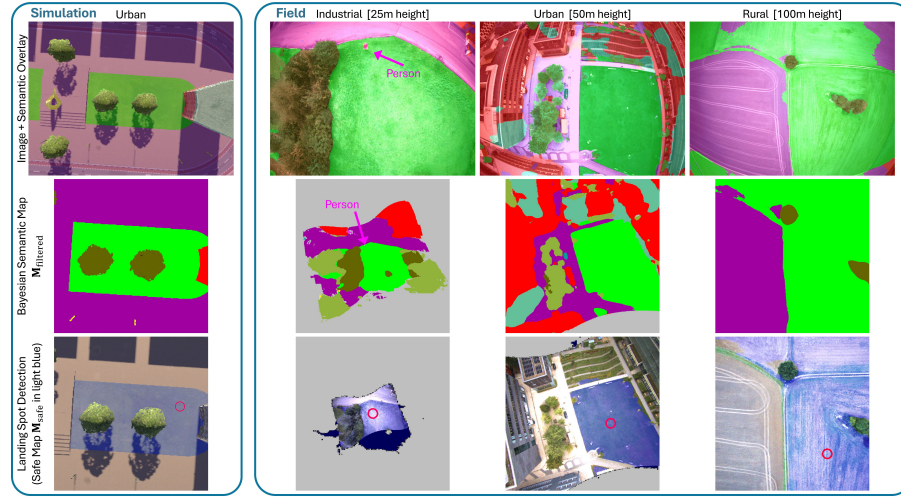


Fig. 6: Experimental results (Sec. 4) in 200 simulations and 60 industrial, urban, and rural field trials (25–100 m AGL) show that SafeLand reliably identifies metric-scale landing sites and responds to non-cooperative humans in real time, achieving 95 % end-to-end landing success, zero false negatives for human detection, and sub-second response latency. Note that the map has a fixed, metric-scale size, resulting in a smaller map at lower heights, as the camera field of view covers less area.

5 Conclusion

In this work, we presented a lightweight vision-based framework for safe autonomous landing in unknown and dynamic environments. By integrating semantic segmentation, Bayesian temporal-probabilistic filtering, and behavior tree-based decision making, the proposed system enables autonomous landing site selection and execution without relying on prior maps, predefined landing hubs, or heavy perception sensors, while explicitly considering human safety. Extensive simulation and real-world evaluations demonstrate the effectiveness of the proposed approach, achieving a 95 % landing success rate, sub-second responses to dynamic obstacles, and zero false negatives in human detection. To facilitate further research in vision-based aerial robotics, we release our segmentation model as a plug-and-play ROS2 package.

Future work will focus on relaxing current assumptions by incorporating terrain-aware uncertainty modeling, improving robustness under challenging environmental conditions, and leveraging recent advances in metric depth estimation and large-scale aerial segmentation, such as OccuFly [20] and SegFly [21], to replace dedicated height sensing and extend semantic understanding beyond the current system capabilities.

References

1. PX4 v1.15 uXRCE-DDS Bridge. https://docs.px4.io/v1.15/en/middleware/uxrce_dds, accessed 2026-02-15
2. ArduSimple: simpleRTK2B Pro (2024), <https://www.ardusimple.com/product/simplertk2b-pro/>, accessed: 2025-09-10
3. Aziz, S., Faheem, R., Bashir, M., Khalid, A., Yasin, A.: Unmanned aerial vehicle emergency landing site identification system using machine vision. *Journal of Image and Graphics* **4**, 36–41 (June 2015)
4. Bektash, O., Pedersen, J.N., Ramirez Gomez, A., Cour-Harbo, A.I.: Automated emergency landing system for drones: Safeeye project. In: 2020 International Conference on Unmanned Aircraft Systems (ICUAS). pp. 1056–1064 (2020). <https://doi.org/10.1109/ICUAS48674.2020.9214073>
5. Bektash, O., Naundrup, J.J., la Cour-Harbo, A.: Analyzing visual imagery for emergency drone landing on unknown environments. *International Journal of Micro Air Vehicles* **14**, 17568293221106492 (2022). <https://doi.org/10.1177/17568293221106492>
6. Bodunkov, N., Kim, N.: Autonomous Landing-Site Selection for a Small Drone. *Russian Engineering Research* **41**, 72–75 (March 2021). <https://doi.org/10.3103/S1068798X2101007X>
7. Cai, W., Jin, K., Hou, J., Guo, C., Wu, L., Yang, W.: Vdd: Varied drone dataset for semantic segmentation (2024), <https://arxiv.org/abs/2305.13608>
8. Carney, E., Castano, L., Xu, H.: Determination of Safe Landing Zones for an Autonomous UAS using Elevation and Population Density Data. *American Institute of Aeronautics and Astronautics SciTech 2019* (January 2019). <https://doi.org/10.2514/6.2019-1060>, <https://arc.aiaa.org/doi/abs/10.2514/6.2019-1060>
9. Chen, J., Du, W., Lin, J., Borhan, U.M., Lin, Y., Du, B., Li, X., Li, J.: Emergency uav landing on unknown field using depth-enhanced graph structure. *IEEE Transactions on Automation Science and Engineering* pp. 1–12 (2024). <https://doi.org/10.1109/TASE.2024.3391017>
10. Chen, Y., Wang, Y., Lu, P., Chen, Y., Wang, G.: Large-scale structure from motion with semantic constraints of aerial images. In: *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. pp. 347–359. Springer (2018)
11. Colledanchise, M., Ögren, P.: Behavior trees in robotics and ai (Jul 2018). <https://doi.org/10.1201/9780429489105>, <http://dx.doi.org/10.1201/9780429489105>
12. De La Torre-Vanegas, J., Soriano-García, M., Becerra, I., Mercado-Ravell, D.: Vision-based risk-aware emergency landing for aerial vehicles in complex urban environments. *Engineering Applications of Artificial Intelligence* **176**, 114823 (2026). <https://doi.org/10.1016/j.engappai.2026.114823>, <http://dx.doi.org/10.1016/j.engappai.2026.114823>
13. Fabbri, R., Costa, L.D.F., Torelli, J.C., Bruno, O.M.: 2d euclidean distance transform algorithms: A comparative survey. *ACM Comput. Surv.* **40**(1) (feb 2008). <https://doi.org/10.1145/1322432.1322434>, <https://doi.org/10.1145/1322432.1322434>
14. Farajijalal, M., Eslamiat, H., Avineni, V., Hettel, E., Lindsay, C.: Safety systems for emergency landing of civilian unmanned aerial vehicles (uavs)—a comprehensive review. *Drones* **9**(2) (2025). <https://doi.org/10.3390/drones9020141>, <https://doi.org/10.3390/drones9020141>
15. FLIR: Blackfly S PGE-31S4C-C. <https://www.flir.com/products/blackfly-s-gige/?model=BFS-PGE-31S4C-C&vertical=machine+vision&segment=iis>, accessed: 2026-02-15

16. Fraczek, P., Mora, A., Kryjak, T.: Embedded Vision System for Automated Drone Landing Site Detection. In: Chmielewski, L.J., Kozera, R., Orłowski, A., Wojciechowski, K., Bruckstein, A.M., Petkov, N. (eds.) *Computer Vision and Graphics*. pp. 397–409. Springer International Publishing, Cham (2018)
17. Gomez, A.R., la Cour-Harbo, A.: Emergency Landing Decision Method for Unmanned Aircraft. In: 2021 International Conference on Unmanned Aircraft Systems (ICUAS). pp. 1233–1239 (2021). <https://doi.org/10.1109/ICUAS51884.2021.9476792>
18. Gonzalez-Trejo, J., Mercado-Ravell, D., Becerra, I., Murrieta-Cid, R.: On the visual-based safe landing of uavs in populated areas: A crucial aspect for urban deployment. *IEEE Robotics and Automation Letters* **6**, 7902–7909 (10 2021). <https://doi.org/10.1109/LRA.2021.3101861>
19. Gonzalez-Trejo, J.A., Mercado-Ravell, D.A.: Lightweight Density Map Architecture for UAVs Safe Landing in Crowded Areas. *Journal of Intelligent & Robotic Systems* **102**(1), 7 (4 2021). <https://doi.org/10.1007/s10846-021-01380-8>, <https://doi.org/10.1007/s10846-021-01380-8>
20. Gross, M., Matha, S.B., Fahmy, A., Song, R., Cremers, D., Meess, H.: OccuFly: A 3D Vision Benchmark for Semantic Scene Completion from the Aerial Perspective. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2026)
21. Gross, M., Matha, S.B., Song, R., Muthuveerappan, V., Christoph, C., Huber, J., Cremers, D.: SegFly: A Dataset and 2D-3D-2D Paradigm for Aerial RGB-Thermal Semantic Segmentation at Scale (2026), <https://arxiv.org/abs/2603.17920>
22. Guerin, J., Delmas, K., Guiochet, J.: Certifying emergency landing for safe urban uav. In: 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). pp. 55–62 (2021). <https://doi.org/10.1109/DSN-W52860.2021.00020>
23. Hausmann, P., Meeß, H., Elger, G.: Image segmentation based emergency landing for autonomous and automated unmanned aerial vehicles. In: 33rd congress of the international council of the aeronautical sciences (September 2022)
24. Hinzmann, T., Stastny, T., Cadena, C., Siegwart, R., Gilitschenski, I.: Free LSD: Prior-Free Visual Landing Site Detection for Autonomous Planes. *IEEE Robotics and Automation Letters* **3**(3) (2018). <https://doi.org/10.1109/LRA.2018.2809962>
25. Institute of Computer Graphics and Vision, TU Graz: Semantic drone dataset, <https://ivc.tugraz.at/research-project/semantic-drone-dataset/>, accessed 09.12.2025
26. Kaljahi, M.A., Shivakumara, P., Idris, M.Y.I., Anisi, M.H., Lu, T., Blumenstein, M., Noor, N.M.: An automatic zone detection system for safe landing of uavs. *Expert Systems with Applications* **122**, 319–333 (2019). <https://doi.org/https://doi.org/10.1016/j.eswa.2019.01.024>, <https://www.sciencedirect.com/science/article/pii/S0957417419300272>
27. Lee, M.F.R., Nugroho, A., Le, T.T., Bahrudin, Bastida, S.N.: Landing Area Recognition using Deep Learning for Unammaned Aerial Vehicles. In: 2020 International Conference on Advanced Robotics and Intelligent Systems (ARIS). pp. 1–6 (2020). <https://doi.org/10.1109/ARIS50834.2020.9205793>
28. Lightware: SF11/C. <https://lightwarelidar.com/shop/sf11-c-100-m/>, accessed: 2026-02-15
29. Loureiro, G., Dias, A., Martins, A., Almeida, J.: Emergency landing spot detection algorithm for unmanned aerial vehicles. *Remote Sensing* **13**(10) (2021). <https://doi.org/10.3390/rs13101930>, <https://www.mdpi.com/2072-4292/13/10/1930>

30. Lusk, P.C., Glaab, P.C., Glaab, L.J., Beard, R.W.: Safe2ditch: Emergency landing for small unmanned aircraft systems. *Journal of Aerospace Information Systems* **16**(8), 327–339 (2019). <https://doi.org/10.2514/1.1010706>, <https://doi.org/10.2514/1.1010706>
31. Lyu, Y., Vosselman, G., Xia, G.S., Yilmaz, A., Yang, M.Y.: Uavid: A semantic segmentation dataset for UAV imagery. *ISPRS Journal of Photogrammetry and Remote Sensing* **165**, 108 – 119 (2020). <https://doi.org/https://doi.org/10.1016/j.isprsjprs.2020.05.009>, <http://www.sciencedirect.com/science/article/pii/S0924271620301295>
32. Macenski, S., Foote, T., Gerkey, B., Lalancette, C., Woodall, W.: Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics* **7**(66), eabm6074 (2022). <https://doi.org/10.1126/scirobotics.abm6074>, <https://www.science.org/doi/abs/10.1126/scirobotics.abm6074>
33. Matsumoto, T., Premachandra, C.: Depth Sensor Application in Ground Unevenness Estimation for UAV Emergency Landing. In: 2023 IEEE Sensors Applications Symposium (SAS). pp. 1–6 (2023). <https://doi.org/10.1109/SAS58821.2023.10253979>
34. Meier, L., Honegger, D., Pollefeys, M.: Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 6235–6240 (2015). <https://doi.org/10.1109/ICRA.2015.7140074>
35. Merei, A., Mcheick, H., Ghaddar, A., Beltrame, G.: Esls: A vision-based emergency safe landing system for uavs. *Procedia Computer Science* **272**, 277–285 (2025). <https://doi.org/https://doi.org/10.1016/j.procs.2025.10.206>, <https://www.sciencedirect.com/science/article/pii/S1877050925035537>, 16th International Conference on Emerging Ubiquitous Systems and Pervasive Networks / 15th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare
36. Modellbau, P.: Quadcopter Q-685. <https://www.premium-modellbau.de/pm-q-685-klappbarer-carbon-quadcopter-rahmen-rtf-komplettsset-frsky> (2024), accessed: 2026-02-15
37. Mukadam, K., Sinh, A., Karani, R.: Detection of landing areas for unmanned aerial vehicles. In: 2016 International Conference on Computing Communication Control and automation (ICCUBE) (2016). <https://doi.org/10.1109/ICCUBE.2016.7860044>
38. Neves, F., Branco, L., Pereira, M., Claro, R., Pinto, A.: A multimodal learning-based approach for autonomous landing of uav (2024), <https://arxiv.org/abs/2405.12681>
39. Neves, F.S., Branco, L.M., Pereira, M.I., Claro, R.M., Pinto, A.M.: A multimodal learning-based approach for autonomous landing of uav. In: 2024 20th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA). pp. 1–8 (2024). <https://doi.org/10.1109/MESA61532.2024.10704866>
40. Nigam, I., Huang, C., Ramanan, D.: Ensemble knowledge transfer for semantic segmentation. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1499–1508 (2018). <https://doi.org/10.1109/WACV.2018.00168>
41. NVIDIA: Jetson AGX Orin Developer Kit. <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin>, accessed: 2026-02-15
42. NVIDIA Corporation: NVIDIA TensorRT. <https://developer.nvidia.com/tensorrt> (2026), accessed: 2026-02-15

43. Osco, L.P., Wu, Q., de Lemos, E.L., Gonçalves, W.N., Ramos, A.P.M., Li, J., Marcato, J.: The segment anything model (sam) for remote sensing applications: From zero to one shot. *International Journal of Applied Earth Observation and Geoinformation* **124** (2023). <https://doi.org/https://doi.org/10.1016/j.jag.2023.103540>, <https://www.sciencedirect.com/science/article/pii/S1569843223003643>
44. Patruno, C., Nitti, M., Petitti, A., Stella, E., D’Orazio, T.: A Vision-Based Approach for Unmanned Aerial Vehicle Landing. *Journal of Intelligent & Robotic Systems* **95** (August 2019). <https://doi.org/10.1007/s10846-018-0933-2>
45. Pieczyński, D., Ptak, B., Kraft, M., Piechocki, M., Aszkowski, P.: A fast, lightweight deep learning vision pipeline for autonomous uav landing support with added robustness. *Engineering Applications of Artificial Intelligence* **131**, 107864 (2024). <https://doi.org/https://doi.org/10.1016/j.engappai.2024.107864>, <https://www.sciencedirect.com/science/article/pii/S0952197624000228>
46. Pixhawk: The Cube Orange. <https://ardupilot.org/copter/docs/common-thecubeorange-overview.html>, accessed: 2026-02-15
47. Rahnmooonfar, M., Chowdhury, T., Sarkar, A., Varshney, D., Yari, M., Murphy, R.R.: Floodnet: A high resolution aerial imagery dataset for post flood scene understanding. *IEEE Access* **9**, 89644–89654 (2021). <https://doi.org/10.1109/ACCESS.2021.3090981>
48. Saldiran, E., Hasanzade, M., Cetin, A., Inalhan, G.: Ensuring operation time safety of vtol uav: Autonomous emergency landings in unknown terrain. *IEEE Transactions on Aerospace and Electronic Systems* **61**(4), 10694–10708 (2025). <https://doi.org/10.1109/TAES.2025.3559899>
49. Semerikov, S.O., Nechypurenko, P.P., Vakaliuk, T.A., Mintii, I.S., Kolhatin, A.O.: Vision-based autonomous UAV landing: A comprehensive review of technologies, techniques, and applications. *Journal of Intelligent and Robotic Systems* **111**(4) (Oct 2025)
50. Sielemann, A., Wolf, S., Roschani, M., Ziehn, J., Beyerer, J.: Synset boulevard: A synthetic image dataset for vmr*. In: 2024 IEEE International Conference on Robotics and Automation (ICRA) (2024). <https://doi.org/10.1109/ICRA57147.2024.10610650>
51. Speth, S., Gonçalves, A., Rigault, B., Suzuki, S., Bouazizi, M., Matsuo, Y., Prendinger, H.: Deep learning with rgb and thermal images onboard a drone for monitoring operations. *Journal of Field Robotics* **39**(6), 840–868 (2022). <https://doi.org/https://doi.org/10.1002/rob.22082>, <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.22082>
52. Tan, Z., He, B., Ji, Y., Wu, L.: Vislanding: Monocular 3d perception for uav safe landing via depth-normal synergy. In: 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 3525–3532 (2025). <https://doi.org/10.1109/IROS60139.2025.11247278>
53. Tovanche-Picon, H., González-Trejo, J., Flores-Abad, Á., García-Terán, M.Á., Mercado-Ravell, D.: Real-time safe validation of autonomous landing in populated areas: from virtual environments to Robot-In-The-Loop. *Virtual Reality* **28**(1) (Mar 2024)
54. Tovanche-Picon, H., Gonzalez-Trejo, J., Flores-Abad, A., Mercado-Ravell, D.: Visual-based safe landing for uavs in populated areas: Real-time validation in virtual environments (2022)
55. Turan, V., Avşar, E., Asadi, D., Aydın, E.A.: Image processing based autonomous landing zone detection for a multi-rotor drone in emergency situations. *Turkish Journal of Engineering* **5**(4) (2021). <https://doi.org/10.31127/tuje.744954>

56. Xia, B., He, T., Mantegh, I., Xie, W.: Ai-based de-confliction and emergency landing algorithm for uas (September 2022). <https://doi.org/10.14339/ST0-MP-AVT-353-12-PDF>
57. Xia, B., Mantegh, I., Xie, W.: Integrated Emergency Self-Landing Method for Autonomous UAS in Urban Aerial Mobility. In: 2021 21st International Conference on Control, Automation and Systems (ICCAS). pp. 275–282 (2021). <https://doi.org/10.23919/ICCAS52745.2021.9649955>
58. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. In: Neural Information Processing Systems (2021)
59. Zhao, C., Kao, X., Lu, Y., Lyu, Y.: A bi-directional adaptive framework for agile uav landing. IEEE Robotics and Automation Letters **11**(5), 5350–5357 (2026). <https://doi.org/10.1109/LRA.2026.3671567>
60. Łuczak, P., Granosik, G.: Autonomous UAV landing and collision avoidance system for unknown terrain utilizing depth camera with actively actuated gimbal. Sensors **25**(19), 6165 (2025). <https://doi.org/10.3390/s25196165>, <https://doi.org/10.3390/s25196165>