Simulation and Analysis of UAV Safe Landing Methods in Urban Scenarios

Marcelo V. Santos
Department of Computer Science
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
marcelo.santos@dcc.ufmg.br

Erickson R. Nascimento
Department of Computer Science
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
erickson@dcc.ufmg.br

***Abstract*—The increasing deployment of unmanned aerial vehicles (UAVs) in urban environments necessitates robust systems to enable safe emergency landings amidst crowded conditions. This study addresses key gaps in existing approaches by developing a comprehensive simulation framework using Unreal Engine and AirSim. The framework allows systematic variation of critical parameters such as crowd density, people speed, lighting, image corruption and data processing latency, providing a robust foundation for evaluating UAV safe landing systems. Safety map generation methods based on crowd density-based maps and people localization maps were analyzed to explore their trade-offs. The density-based approach demonstrated superior robustness to motion blur, maintaining usable performance under moderate corruption levels. However, it faced challenges with higher abort rates and longer mission durations. The localization-based method had better performance in normal visual conditions, achieving higher success rates across varying crowd densities, speeds and data processing latencies. This method also showed faster mission completion times in all experiments. We conclude that the density-based method is more suitable for harsh visual conditions, while the localization-based method is preferable for normal conditions or when timing is critical. Future work should focus on hybrid methods that dynamically adapt to the environment, real-world validation and alignment with regulatory standards to ensure deployment readiness. These advancements will enable safer UAV operations in urban settings.**

***Index Terms*—Autonomous UAV Safe Landing, Robust UAV Systems, Urban Environments, Crowd Density, Crowd Localization, Distance Map, Simulation, AirSim**

I. Introduction

Unmanned Aerial Vehicles (UAVs) have become increasingly valuable across numerous economic sectors, with their global market valued at USD 27.43 billion in 2022 and projected to reach USD 91.23 billion by 2030, demonstrating a compound annual growth rate of 16.3% [1]. As of 2025, the Federal Aviation Administration had registered 855,860 drones in the United States alone, reflecting their widespread adoption, which spans fields such as defense, disaster response, crowd monitoring, surveillance, logistics, agriculture, construction, and media production [1]–[3].

However, operating these systems in populated areas introduces serious safety issues, particularly as their applications expand into urban environments. Failures caused by loss of communication, low battery, or adverse weather conditions can result in unsafe landings, posing significant risks to human life and property. Current UAV regulations prioritize risk avoidance, with agencies like the FAA and EASA imposing strict restrictions in populated areas [4], [5]. However, completely avoiding people in urban settings is impractical. This inability to guarantee safe landings limits UAV adoption, especially for applications like urban delivery [6]. Autonomous emergency landing systems could improve safety and enable more flexible regulations [7], but technical challenges in perception, decision-making and validation are significant.

Urban settings are dense with obstacles, complicating real-time perception. Onboard sensors must detect potential hazards with high reliability despite occlusions, lighting variability and possible hardware degradation during emergencies. While deep learning-based models have shown promise in controlled conditions [8]–[10], real-world reliability under uncertainty remains an open problem.

Beyond perception, the UAV must rapidly evaluate and commit to a viable landing site under time and control constraints. Urban settings further complicate this process with unpredictable pedestrian and vehicle movement, therefore, the UAV's decision algorithm must consider short-term future conditions and include abort maneuvers, introducing complexity into the system.

Besides, regulatory approval will require strong, verifiable evidence of system reliability, but proving that the system will perform reliably in diverse real world scenarios is non-trivial. Current testing heavily relies on simulation, but simulations can overlook real-world sensor noise and rare events. Much of the current work is still experimental and not directly comparable due to different assumptions [7], making it hard to stablish performance benchmarks for safe landing systems.

In this context, recent research in computer vision and robotics has advanced vision-based safe landing systems for UAVs, particularly through crowd detection. Initial efforts [8]–[11] lacked realistic validation, which was later addressed using AirSim [12] and Unreal Engine. Kakaletsis et al. [13] tested in simulated rural settings, while Gonzalez-Trejo et al. [14], [15] focused on simulated urban environments. However, these studies varied few experimental parameters and within narrow ranges.

Our work addresses these limitations by introducing a more comprehensive simulation framework that evaluates safe

landing methods under diverse and challenging conditions. Unlike previous efforts, our framework controls and systematically varies multiple parameters, such as lighting level, crowd density and people speed. It also accounts for image corruption and image processing latency. By testing both dilated density maps, as used by Tovanche-Picon et al. [15], [16], and people localization maps as inputs to the UAV safe landing algorithm, this study also explores the impact of different people localization methods on overall safe landing performance. Our approach complements previous research by expanding the range of scenarios considered and sets a new standard for simulation-based validation of UAV safe landing systems in densely populated urban environments.

II. Related Work

A. Safe Landing Zone Detection

Early methods for UAV landing site detection relied on handcrafted features and traditional computer vision techniques. The first works [17], [18] focused on detecting known visual markers to avoid landing on terrain, while others tried to detect safe landing zones based on classical image features such as Histogram of Oriented Gradients [19], [20]. Beyond image classification, researchers have also explored 3D reconstruction techniques to gain a more comprehensive understanding of the landing environment [21], [22].

Other approaches use deep neural networks (DNNs), which offer better accuracy. Marcu et al. [23] introduced two CNN models that segment aerial images into horizontal, vertical and undefined regions, treating horizontal zones as landing candidates. Horizontal areas are considered potential landing zones. In contrast, Polvara et al. [24] proposed a deep reinforcement learning approach that maps images directly to control actions using a deep Q-network.

While prior methods improve landing zone detection, they do not address the critical risk of harming individuals, which is essential for safety and regulatory compliance.

B. Crowd Detection for UAV Safe Landing

Tzelepi et al. [8], [9] developed lightweight CNN models for UAV crowd detection. However, their methods focus on detection and were not integrated into a safe landing pipeline to be evaluated in landing simulations. More recent work by Kakaletsis et al. [13] incorporated crowd detection into a full safe landing pipeline. They proposed a modular pipeline for UAV safe landing, combining segmentation, detection and planning components. Although they evaluated the method in simulation, it was tested in only 15 missions and focused on sparsely populated environments, limiting its applicability to dense urban scenarios addressed in subsequent work.

Gonzalez-Trejo and Mercado-Ravell [10] developed a lightweight CNN for crowd density estimation. Named Pruned BL CCNN, it builds upon the Compact Convolutional Neural Network [25] and uses a specialized loss function, the Bayes Loss (BL) [26]. Traditional crowd counting methods often use the Euclidean distance as the loss function to compare predicted density maps with ground truth. However, this approach does not account for slight inaccuracies in annotation placement, such as labels not being perfectly centered on a person's head, and it ignores the spatial relationships between neighboring pixels. The Bayes Loss addresses these issues by reframing the problem probabilistically, treating the ground truth density maps as likelihoods rather than fixed targets and operating per label instead of per pixel. This improves the robustness of crowd counting models, achieving superior performance over traditional approaches on benchmark datasets [26].

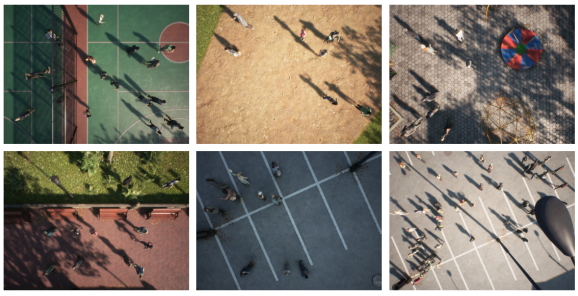
Building on this, Gonzalez-Trejo et al. [10] proposed a density map-based pipeline for detecting safe landing zones (SLZs) in crowded environments. The method identifies candidate sites and selects the safest using geometric criteria. Subsequent work improved SLZ stability in dynamic settings through Kalman filtering [14] and validated the system using simulations [15]. The method was tested under a diverse range of conditions in urban scenarios populated with randomized crowd distributions, weather effects and actor movements [15]. The approach was later extended to Robot-In-The-Loop (RITL) testing to combine real hardware with virtual environments [16].

III. Methodology

A. City Park Dataset

The City Park dataset was developed to train the crowd localization models used in this work. Since the evaluation occurs in a synthetic environment, the dataset was created to ensure that the trained models operate within the same domain. Generated using Unreal Engine 4.27 and the City Park LITE environment, it simulates urban park settings. The dataset comprises 3,000 training and 600 test images at 768×576 resolution, covering varied locations for both training (a basketball court, a baseball field, a children's park, a city square and a parking lot) and testing (green terrain, a soccer field and a road for testing). Figure 1 shows images from the training scenarios.

Fig. 1. Examples of training images from the City Park dataset. The images show various urban scenarios: a basketball court, a baseball field, a playground, a city square and a parking lot.



Training and testing datasets use separate pools of randomized virtual actors: 99 for training and 94 for testing. Actor placement and orientation vary per scene to simulate different crowd patterns, with up to 120 actors per image and

densities reaching 0.5 people/m². Each image has associated head annotations.

Scene lighting is also randomized. In Unreal, it's controleed by setting the intensity scale of the environment's SkyLight object and setting the intensity of the environment's Directional Light object. The Directional Light intensity is sampled from a uniform distribution between 0 and 5 lux and the SkyLight scale is fixed at 0.25. Even at 0 lux, scenes remain illuminated due to the SkyLight. The camera maintains a fixed 90-degree horizontal FOV and a vertical, top-down view.

B. Safety Map Generation

The safety map is a key element of the safe landing algorithm, assigning a safety index to each grid point based on visual input, with higher values indicating safer landing zones. In this work, safety is evaluated using distance to people.

Two generation methods are compared: one based on crowd density estimates and another based on localized detections of individuals. Both rely on a fixed head plane, which represents the estimated height of people's heads above the ground, to convert pixel measurements into real-world distances.

1) Crowd Density Map-Based Safety Map Generation: For this method, we use the crowd density estimation model from [26], a CNN trained with Bayes loss to produce density maps at $1/8$ the input resolution. Each pixel represents the estimated local crowd density. The model was trained using standard hyperparameters and validated on a 600-image validation subset of the training data. The best-performing checkpoint, based on Mean Absolute Error (MAE) and Mean Squared Error (MSE), was selected. It achieved MAE of 1.972 and MSE of 3.183 for crowd counting with the testing data.

The process of generating a safety map from a crowd density map was based in the method presented in [10]. First, a binary mask is generated by thresholding the crowd density map. Pixels with a value greater than 1×10^{-4} are set to 255 and all other pixels are set to 0. Then, the binary mask undergoes morphological dilation using a 5×5 kernel. The binary mask is then resized with nearest-neighbor interpolation to match the original image dimensions. The next step is applying the `findContours` function from Open CV [27] to the dilated binary mask to extract the boundaries of occupied regions. These contours are then used to create a "people-free mask," which fills in the identified occupied regions while leaving the remaining areas marked as safe for landing. Finally, an Euclidean distance transform is applied to the people-free mask to compute the distance of each safe pixel to the nearest occupied pixel. These pixel distances are scaled to real-world units using the UAV's distance to the head plane height and the UAV camera's intrinsic parameters. So, the distance to the nearest occupied pixel is the safety index. If the binary mask is completely empty, the center of the image is marked as the safest location, and it is assigned the highest safety score. Figure 2 shows an input image, the density map, the binary mask, the dilated binary mask, the extracted contours and the final safety map generated by this method.

2) People Localization Map-Based Safety Map Generation: The crowd localization network from [28] is used to predict a Focal Inverse Distance Transform (FIDT) map, where each pixel encodes an inverse distance centered around annotated head locations. Before training, point-level head annotations were converted into FIDT maps. The Independent Structural Similarity Loss [28] was used for training.

The model was trained for 400 epochs using the Adam optimizer with a learning rate of 1×10^{-4}, weight decay of 5×10^{-4}, and a batch size of 16. A validation set of 600 training images was used to monitor MAE and MSE every 10 epochs, with the best-performing checkpoint selected to be used in the safe landing algorithm. It achieved a MAE of 0.528 and a MSE of 1.310 for crowd counting with the testing data.

The people localization map is generated from the FIDT map using Local-Maxima Detection Strategy [28]. Then, each pixel in the safety map is assigned a value corresponding to its distance to the nearest occupied pixel. The pixel distances are then converted into real-world distances in the head plane using the UAV's height relative to the head plane and the intrinsic parameters of the UAV camera. Figure 3 shows an input image, the people localization map, and the safety map generated by the people localization map-based approach.

C. UAV Safe Landing Algorithm

The UAV safe landing algorithm is built around the safety map and includes Safety-Guided Navigation and Safety-Guided Landing. It executes until the UAV successfully completes a landing, a collision with a person is detected during descent or the elapsed simulation time exceeds the defined mission timeout. The algorithm enforces a mission timeout of three minutes to prevent indefinite operation.

1) Safety-Guided Navigation: In each iteration, the UAV collects its height and a 768×576 top-down RGB image. The simulation is paused during data acquisition and safety map generation to guarantee that the height and image are synchronized in time, as recommended by AirSim best practices [29].

The data is processed to generate the safety map and crowd map. The safety map is analyzed to identify maximum safety index location and its real-world coordinates in the head plane. To improve reliability, the borders of the safety map are set to zero before evaluating safety indices. The borders are less reliable because the distances do not account for people who may be just outside the UAV's field of view. The size of the ignored border corresponds to a 1.0-meter distance in the real world, capped at a maximum of one-sixth of the image width to prevent excessive masking.

Once the maps are generated, the simulation is unpaused and a wait period is introduced to emulate the latency. This represents time required for data acquisition and processing. These delays are simulated for controlled experimentation and reproducibility. Then, the method checks for mission timeout and evaluates its next action.

If the maximum safety index exceeds a predefined threshold or the mission timeout is reached, the UAV initiates landing. If no person's head is detected in the crowd map or mission

Fig. 2. (a) Input RGB image showing people from a UAV perspective; (b) The predicted crowd density map; (c) Binary mask obtained by thresholding the density map to segment occupied region; (d) Dilated binary mask using a 5×5 kernel to expand unsafe areas; (e) Contours extracted using OpenCV's `findContours` function, outlining people-occupied regions; (f) Final safety map, where each pixel encodes the Euclidean distance to the nearest unsafe region and is scaled to real-world units based on the UAV's height and camera parameters.

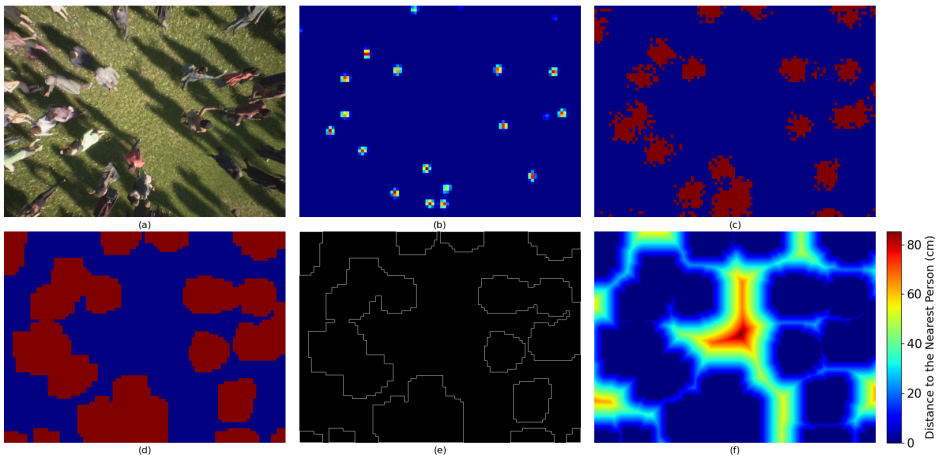
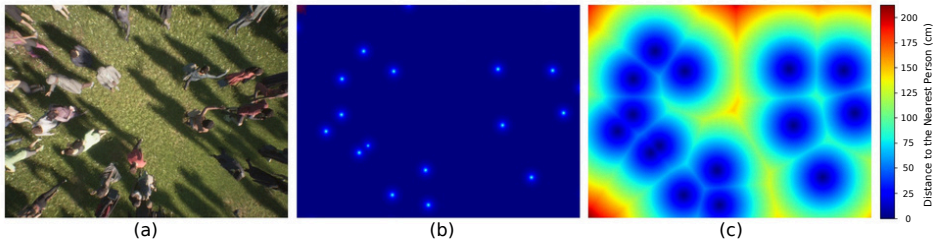


Fig. 3. (a) Input RGB image; (b) People localization map derived via Local-Maxima Detection Strategy (LMDS), where bright peaks correspond to detected head positions extracted from the FIDT map; (c) Final safety map, where each pixel encodes the Euclidean distance to the nearest detected person and is scaled to real-world units based on the UAV's height and camera parameters.



timeout is reached, the UAV performs an immediate landing, descending directly to the safest spot without further assessments. Otherwise, a safety-guided landing is performed. The landing routines are responsible for determining whether the landing has completed successfully or if a collision with a person occurred.

If the maximum safety index does not exceed the threshold and the mission timeout is not reached, the UAV repositions itself. If the safest point is at least $0.1\,\mathrm{m}$ from the UAV's current position, that location is selected as the repositioning target. Otherwise, the UAV selects the corner of its field of view with the fewest detected people, as inferred from the crowd map. This repositioning logic addresses an issue in AirSim, where commanding the UAV to move to a nearby point can cause sudden descents. Repositioning is executed as a horizontal movement over $0.667\,\mathrm{s}$. The UAV maintains a constant height throughout the motion, as specified at the beginning of the mission.

An empirically determined minimum interval of two seconds is enforced between successive iterations of the Safety-Guided Navigation phase. This avoids rapid successive control commands, which can lead to erratic UAV behavior in the AirSim simulation environment.

2) Safety-Guided Landing: Once a safe location is identified, the UAV initiates controlled landing. Data acquisition and safety map generation during Safety-Guided Landing follow the same methodology as in the Safety-Guided Navigation phase.

The UAV initiates a descent path with two waypoints: one directly above the landing location and the other at the target coordinates at ground level. The checkpoint is placed at a buffer height of 1.0 meter above the head plane. In our simulations, the head plane height is fixed at 2.0 meters.

Until the checkpoint is reached, the UAV monitors the safety index of the designated landing zone. If the safety index falls below the threshold, the UAV generates a new safety map to search for an alternative. If safe location is identified, the UAV adjusts its trajectory to target it. The new descent path is constructed in the same manner as the previous one.

After reaching the checkpoint, depending on the safety map generation method, either the crowd density map or the people localization map is analyzed to detect whether any individuals have entered the landing zone. If any actors are detected within the UAV's field of view, it aborts the landing.

To abort a landing, the UAV repositions itself to the predefined mission height, directly above its current location. It then resumes the Safety-Guided Navigation phase.

The mission is terminated and marked as a failure if the UAV collides with a person during descent. The descent is successful if the UAV reaches an altitude of 0.4 meters above the ground plane without incidents. This slightly elevated height was chosen to circumvent a bug in AirSim, where the UAV failing to recognize ground contact prevented mission termination.

D. Simulation

The methods were evaluated in a simulated environment built with Unreal Engine 4.27 and the City Park Environment Collection LITE. The setup uses the scenarios and actors from the City Park test dataset. The UAV camera features a 90° horizontal field of view, a resolution of 768×576 pixels and a downward-facing gimbal-stabilized orientation for top-down image capture.

The crowd movement in the simulation is randomized. Actors are assigned a maximum speed $v_{\max}$, defining the maximum one-second travel distance $d_{\max}$. A random target is selected within a square centered on the actor's position, with a diagonal of $2 \cdot d_{\max}$. The actor moves toward the target with the speed to reach it in one second and with position updates every 0.1 seconds. Then, the process repeats from the new position.

The simulation uses the following default parameters: 2.5 lux light intensity, 0.5 people/m^2 density, 1.4 m/s maximum speed and a UAV altitude of 5.0 m. The head plane height is set to 2.0 m. Minimum safety distances are 0.9 m for the crowd density method and 2.0 m for the people localization method. A 0.1-second latency models the time for image acquisition and safety map generation. Minimum safety distance values were selected based on experiments discussed in the next section.

Each experiment configuration included 15 missions per scenario, totaling 45 runs. The following metrics were averaged: mission success (no collisions with humans), time to complete the mission, number of aborted landings and the UAV's final distance to the nearest person.

IV. Results and Discussion

In this section, we present the experiments conducted and compare results for the two safety map generation methods. Experiments are grouped into algorithm configuration, crowd dynamics and visual conditions.

A. Algorithm Configuration

1) Safety Index Threshold:

a) Threshold Analysis for Safety Map Generation from Crowd Density Map: To tune the minimum safety threshold for density-based safety maps, thresholds from 0.0 m to 1.0 m were tested (Table I). Success rates increased from 31.11% at 0.0 m to 82.22% at 0.9 m, with a slight drop at 1.0 m. Higher thresholds also resulted in longer missions and more aborted landings, peaking at 120.02 s and 22.13 aborts at 0.9 m. Nearest person distance improved accordingly, reaching 2.06 m at 0.9 m. Based on these results, 0.9 m was selected for subsequent experiments to prioritize safety.

b) Threshold Analysis for Safety Map Generation from People Localization Map: To tune the minimum safety threshold for localization-based safety maps, thresholds from 1.0 m to 3.0 m were tested (Table II). Success rates peaked at 82.22% at 2.0 m, with longer mission durations and increased nearest person distances at higher thresholds. Abort rates were highest at 2.0 m, reflecting more active landing attempts. Based on a balance of safety and speed, 2.0 m was selected for future experiments.

c) Method Comparison: The localization-based safety map method consistently outperforms the crowd density-based approach across key metrics. While both methods achieve their highest success rate of 82.22 % (at 0.9 m and 2.0 m respectively), localization-based maps maintain higher success rates across all thresholds. Mission durations are lower for density-based maps at minimal thresholds, but localization-based maps achieve their peak success with a 34.58 % shorter mission time compared to the density-based method. Abort rates are also lower for localization-based maps, indicating more cautious and reliable landing behavior. However, density-based maps achieved greater nearest person distances in the optimal configuration.

2) Data Acquisition and Safety Map Generation Latency: As shown in Table III, increasing latency degrades performance in both methods, but the localization-based approach is more robust. Its success rate declines gradually from 82.22 % at 100 ms to 57.78 % at 500 ms, while the density-based method drops more sharply to 46.67 %. Abort rates are consistently lower for the localization-based method, indicating better stability. Nearest person distances are slightly better for the density-based method. The localization-based method maintains acceptable performance up to 400 ms.

B. Crowd Dynamics

1) People Density: As shown in Table IV, both methods perform well under low crowd densities, achieving over 97 % success at 0.1–0.2 people/m². As density increases, the density-based method achieves higher success at 0.4 people/m² and matches the localization-based method at 0.5 people/m². The localization-based method consistently completes missions faster and exhibits lower abort rates across all densities. At 0.5 people/m², the density-based method records 22.13 aborted landings, compared to 5.78 for the localization-based method. Nearest person distances are generally higher for the density-based method. Overall, the density-based method offers better safety margins, but with increased mission time and abort frequency in dense environments.

2) Maximum People Speed: As shown in Table V, both methods show reduced performance as pedestrian speed increases. The localization-based method generally achieves higher success rates and lower abort rates across all speeds. Its highest success occurs at 0.7 m/s (97.78%), while the density-based method performs best with stationary people (100.00 % at 0.0 m/s). At the highest tested speed (2.8 m/s), success drops to approximately 50% for both methods. Abort rates for the density-based method rise sharply, peaking at 24.31, while the

TABLE I

LANDING SUCCESS RATE, TIME TO COMPLETE, NUMBER OF LANDINGS ABORTED AND NEAREST PERSON DISTANCE FOR DIFFERENT SAFETY THRESHOLDS IN DENSITY-BASED SAFETY MAPS.

Safety Threshold (m)	**Success Rate (%)**	**Time to Complete (s)**	**Landings Aborted**	**Nearest Person Distance** (m)
0.0	31.11	6.30	1.00	0.75
0.5	40.00	12.01	1.91	0.87
0.6	48.89	18.71	3.20	0.96
0.7	68.89	33.51	6.42	1.22
0.8	68.89	63.57	12.22	1.41
0.9	82.22	120.02	22.13	2.06
1.0	77.78	117.58	17.87	1.87

TABLE II

LANDING SUCCESS RATE, TIME TO COMPLETE, NUMBER OF LANDINGS ABORTED AND NEAREST PERSON DISTANCE FOR DIFFERENT SAFETY THRESHOLDS IN PEOPLE LOCALIZATION-BASED SAFETY MAPS.

Safety Threshold (m)	**Success Rate (%)**	**Time to Complete (s)**	**Landings Aborted**	**Nearest Person Distance** (m)
1.0	55.56	8.84	1.13	0.99
1.5	62.22	24.16	3.98	1.15
2.0	82.22	78.52	5.78	1.86
2.5	68.89	130.73	3.04	2.23
3.0	73.33	148.84	1.11	2.41

TABLE III

LANDING SUCCESS RATE, TIME TO COMPLETE, NUMBER OF LANDINGS ABORTED AND NEAREST PERSON DISTANCE FOR DIFFERENT LATENCIES.

Latency (ms)	**Success Rate (%)**		**Time to Complete (s)**		**Landings Aborted**		**Nearest Person Distance** (m)	
	Density	**Localization**	**Density**	**Localization**	**Density**	**Localization**	**Density**	**Localization**
100	82.22	82.22	120.02	78.52	22.13	5.78	2.06	1.86
200	68.89	73.33	72.03	63.17	11.91	4.11	1.55	1.60
300	60.00	77.78	45.53	53.12	6.80	3.09	1.18	1.34
400	44.44	80.00	24.83	61.40	3.44	3.73	0.96	1.38
500	46.67	57.78	25.09	69.63	3.36	4.16	1.01	1.28

TABLE IV

LANDING SUCCESS RATE, TIME TO COMPLETE, NUMBER OF LANDINGS ABORTED, AND NEAREST PERSON DISTANCE FOR DIFFERENT PEOPLE DENSITIES FOR BOTH METHODS.

People Density (m^{-2})	**Success Rate (%)**		**Time to Complete (s)**		**Landings Aborted**		**Nearest Person Distance** (m)	
	Density	**Localization**	**Density**	**Localization**	**Density**	**Localization**	**Density**	**Localization**
0.1	97.78	100.00	9.16	7.09	0.93	0.56	2.42	2.26
0.2	100.00	97.78	18.76	10.61	2.89	1.33	2.04	1.86
0.3	93.33	93.33	47.85	18.32	9.11	2.44	1.73	1.74
0.4	86.67	77.78	80.76	43.68	16.13	4.58	1.64	1.47
0.5	82.22	82.22	120.02	78.52	22.13	5.78	2.06	1.86

localization-based method remains significantly more stable. Mission duration is consistently shorter for the localization-based method.

C. Visual Conditions

The experiments in this section, except the one that varied lighting level, were conducted by adding noise to the images before generating the safety map. It was added with the *imagecorruptions* Python library [30]. The remaining experiment was conducted by changing the lighting level in the simulation environment.

1) Motion Blur: Table VI shows that motion blur significantly degrades performance for both methods, though the crowd density-based method consistently outperforms the localization-based method. Success rates for the density-based approach decline from 82.22% with no blur to 11.11% at level 5. The localization-based method deteriorates more rapidly, dropping to 40.00 % at level 2 and 22.22 % at level 5. Abort rates and mission durations decrease at higher blur levels, reflecting fewer landing attempts and early mission terminations. Nearest person distances also shrink across both methods, though the density-based method maintains slightly better safety margins except for level 5.

2) Lighting: Table VII shows that both methods maintain high and stable performance across varying lighting conditions. The crowd density-based method achieves its peak suc-

TABLE V
LANDING SUCCESS RATE, TIME TO COMPLETE, NUMBER OF LANDINGS ABORTED, AND NEAREST PERSON DISTANCE FOR VARYING PEOPLE SPEEDS FOR BOTH METHODS.

People Speed (m/s)	**Success Rate (%)**		**Time to Complete (s)**		**Landings Aborted**		**Nearest Person Distance** (m)	
	Density	**Localization**	**Density**	**Localization**	**Density**	**Localization**	**Density**	**Localization**
0.0	100.00	95.56	140.49	88.77	26.16	2.44	1.77	1.98
0.7	86.67	97.78	118.25	71.01	21.13	3.18	1.95	2.00
1.4	82.22	82.22	120.02	78.52	22.13	5.78	2.06	1.86
2.1	55.56	60.00	122.61	72.09	24.31	5.44	1.53	1.60
2.8	48.89	51.11	117.91	85.31	22.24	6.69	1.81	1.44

TABLE VI
LANDING SUCCESS RATE, TIME TO COMPLETE, NUMBER OF LANDINGS ABORTED, AND NEAREST PERSON DISTANCE FOR VARYING MOTION BLUR INTENSITY.

Motion Blur Intensity Level	**Success Rate (%)**		**Time to Complete (s)**		**Landings Aborted**		**Nearest Person Distance** (m)	
	Density	**Localization**	**Density**	**Localization**	**Density**	**Localization**	**Density**	**Localization**
0	82.22	82.22	120.02	78.52	22.13	5.78	2.06	1.86
1	73.33	68.89	66.79	56.72	12.89	10.89	1.36	1.31
2	68.89	40.00	31.15	15.26	6.16	2.53	1.15	0.89
3	11.11	17.78	5.08	3.97	0.29	0.07	0.68	0.63
4	22.22	24.44	3.75	3.67	0.00	0.00	0.71	0.68
5	11.11	22.22	3.68	3.65	0.00	0.00	0.62	0.68

cess rate of 82.22% at 2.5 lux, while the people localization-based method varies between 71.11% and 84.44%. Mission times for the localization-based method are consistently shorter, with a maximum of 81.34 seconds at 1.25 lux, compared to 126.54 seconds for the density-based method at 5.0 lux. Abort rates are relatively stable across light levels. The density-based method maintains higher abort counts (up to 23.91), while the localization-based method remains below 6.5. Nearest person distances exceed 1.4 meters in all cases, with the density-based method slightly outperforming at some intensities. Overall, lighting variation has minimal impact on UAV safe landing performance.

V. CONCLUSION

Our simulation platform addressed gaps in prior research, providing a robust basis for evaluating and comparing UAV safe landing methods. Two safety map generation methods were compared under varying thresholds and latencies. Both achieved a peak success rate of 82.22%, at 0.9 m and 2.0 m respectively, but the localization-based method maintained better performance under latency up to 400 ms.

The density-based approach proved more resilient to visual corruption but incurred longer missions and higher abort rates. In contrast, the localization-based method performed better under clean conditions and completed missions faster. Therefore, density-based maps are better for harsh environments, while localization-based maps are preferable for time-sensitive conditions.

Despite using high-fidelity simulation, the framework overlooks key real-world complexities such as social crowd behavior, uneven terrain, sensor noise and asynchronous onboard processing. Beyond this, both safety map methods degrade under visual corruption, with the localization-based approach especially vulnerable. The system also presumes access to high-performance GPUs, ignoring constraints of embedded UAVs that may impact latency and safety. Finally, real-world deployment must also meet FAA/EASA regulations and address ethical issues like privacy and fairness, none of which were audited in this work.

Future work could address the system's vulnerability to visual corruption via corruption-aware training, denoising modules and multi-modal sensing. Real-world deployment is also essential: porting the pipeline to embedded UAVs and conducting urban field trials will expose practical limitations. Fairness audits and bias mitigation using fairness-constrained training [31] are an option, along with tools to support interpretability in high-stakes applications. Finally, the simulation platform can be used to evaluate alternate solutions, such as replacing heuristic rules by safety-aware reinforcement learning techniques like constrained policy optimization [32].

Together, these directions aim to evolve the current system from a simulation-validated framework into a robust and trustworthy UAV safe landing solution suitable for real-world applications in urban environments.

REFERENCES

[1] F. B. Insights, *Unmanned Aerial Vehicle (UAV) Market*, 2025, accessed: 2025-3-4. [Online]. Available: https://www.fortunebusinessinsights.com/industry-reports/unmanned-aerial-vehicle-uav-market-101603

[2] J. Leslie, *US Drone Statistics 2025*, 2025, accessed: 2025-3-4. [Online]. Available: https://skykam.co.uk/drone-statistics/

[3] M. C. Horowitz, "Do emerging military technologies matter for international politics?" *Annual Review of Political Science*, vol. 23, no. Volume 23, 2020, pp. 385–400, 2020.

[4] FAA, "Part 107 - small unmanned aircraft systems," 2016, accessed on 30/11/2024. [Online]. Available: https://www.ecfr.gov/current/title-14/chapter-I/subchapter-F/part-107

TABLE VII
LANDING SUCCESS RATE, TIME TO COMPLETE, NUMBER OF LANDINGS ABORTED, AND NEAREST PERSON DISTANCE FOR DIFFERENT LIGHT INTENSITIES.

Light Intensity (lux)	**Success Rate (%)**		**Time to Complete (s)**		**Landings Aborted**		**Nearest Person Distance (m)**	
	Density	**Localization**	**Density**	**Localization**	**Density**	**Localization**	**Density**	**Localization**
0.0	80.00	84.44	73.30	66.38	13.93	6.29	1.47	1.55
1.25	73.33	71.11	107.71	81.34	20.22	6.09	1.58	1.48
2.5	82.22	82.22	120.02	78.52	22.13	5.78	2.06	1.86
3.75	71.11	77.78	108.58	74.14	20.20	5.89	2.17	1.86
5.0	75.56	77.78	126.54	69.09	23.91	5.87	1.59	1.73

[5] JARUS, "Specific operations risk assessment (sora)," 2024, accessed on 1/12/2024. [Online]. Available: http://jarus-rpas.org/wp-content/uploads/2024/06/SORA-v2.5-Main-Body-Release-JAR_doc_25.pdf

[6] J. Guerin, K. Delmas, and J. Guiochet, "Certifying Emergency Landing for Safe Urban UAV," in *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. Taipei, Taiwan: IEEE, Jun. 2021, pp. 55–62.

[7] M. Farajijalal, H. Eslamiat, V. Avineni, E. Hettel, and C. Lindsay, "Safety systems for emergency landing of civilian unmanned aerial vehicles (uavs)—a comprehensive review," *Drones*, vol. 9, no. 141, 2025.

[8] M. Tzelepi and A. Tefas, "Human crowd detection for drone flight safety using convolutional neural networks," in *2017 25th European Signal Processing Conference (EUSIPCO)*. Kos, Greece: IEEE, Aug. 2017, pp. 743–747.

[9] ——, "Graph Embedded Convolutional Neural Networks in Human Crowd Detection for Drone Flight Safety," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 2, pp. 191–204, Apr. 2021.

[10] J. A. Gonzalez-Trejo and D. A. Mercado-Ravell, "Lightweight Density Map Architecture for UAVs Safe Landing in Crowded Areas," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 1, p. 7, May 2021.

[11] H. C. Shadakshri V, V. M. B, and K. R. G. D. V, "OpenCV Implementation of Grid-based Vertical Safe Landing for UAV using YOLOv5," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 9, 2022.

[12] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[13] E. Kakaletsis, C. Symeonidis, M. Tzelepi, I. Mademlis, A. Tefas, N. Nikolaidis, and I. Pitas, "Computer Vision for Autonomous UAV Flight Safety: An Overview and a Vision-based Safe Landing Pipeline Example," *ACM Computing Surveys*, vol. 54, no. 9, pp. 1–37, Dec. 2022.

[14] J. González-Trejo, D. Mercado-Ravell, I. Becerra, and R. Murrieta-Cid, "On the Visual-based Safe Landing of UAVs in Populated Areas: A Crucial Aspect for Urban Deployment," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7901–7908, Oct. 2021.

[15] H. Tovanche-Picon, J. Gonzalez-Trejo, A. Flores-Abad, and D. Mercado-Ravell, "Visual-based Safe Landing for UAVs in Populated Areas: Real-time Validation in Virtual Environments," *Virtual Reality*, vol. 28, no. 1, p. 66, Mar. 2022.

[16] H. Tovanche-Picon, J. González-Trejo, Á. Flores-Abad, M. Á. García-Terán, and D. Mercado-Ravell, "Real-time safe validation of autonomous landing in populated areas: From virtual environments to Robot-In-The-Loop," *Virtual Reality*, vol. 28, no. 1, p. 66, Mar. 2024.

[17] S. Saripalli, J. Montgomery, and G. Sukhatme, "Vision-based autonomous landing of an unmanned aerial vehicle," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 3, 2002, pp. 2799–2804 vol.3.

[18] A. Johnson, J. Montgomery, and L. Matthies, "Vision Guided Landing of an Autonomous Helicopter in Hazardous Terrain," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. Barcelona, Spain: IEEE, 2005, pp. 3966–3971.

[19] X. Guo, S. Denman, C. Fookes, L. Mejias, and S. Sridharan, "Automatic UAV Forced Landing Site Detection Using Machine Learning," in *2014 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. Wollongong, New South Wales, Australia: IEEE, Nov. 2014, pp. 1–7.

[20] X. Guo, S. Denman, C. Fookes, and S. Sridharan, "A robust UAV landing site detection system using mid-level discriminative patches," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. Cancun: IEEE, Dec. 2016, pp. 1659–1664.

[21] V. Desaraju, N. Michael, M. Humenberger, R. Brockers, S. Weiss, and L. Matthies, "Vision-based Landing Site Evaluation and Trajectory Generation Toward Rooftop Landing," in *Robotics: Science and Systems X*. Robotics: Science and Systems Foundation, Jul. 2014.

[22] T. Hinzmann, T. Stastny, C. Cadena, R. Siegwart, and I. Gilitschenski, "Free LSD: Prior-Free Visual Landing Site Detection for Autonomous Planes," Feb. 2018.

[23] A. Marcu, D. Costea, V. Licăreţ, M. Pîrvu, E. Sluşanschi, and M. Leordeanu, "SafeUAV: Learning to Estimate Depth and Safe Landing Areas for UAVs from Synthetic Data," in *Computer Vision – ECCV 2018 Workshops*, L. Leal-Taixé and S. Roth, Eds. Cham: Springer International Publishing, 2019, vol. 11130, pp. 43–58.

[24] R. Polvara, M. Patacchiola, S. Sharma, J. Wan, A. Manning, R. Sutton, and A. Cangelosi, "Toward End-to-End Control for UAV Autonomous Landing via Deep Reinforcement Learning," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. Dallas, TX: IEEE, Jun. 2018, pp. 115–123.

[25] X. Shi, X. Li, C. Wu, S. Kong, J. Yang, and L. He, "A real-time deep network for crowd counting," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 2328–2332.

[26] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Bayesian loss for crowd count estimation with point supervision," 2019. [Online]. Available: https://arxiv.org/abs/1908.03684

[27] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[28] D. Liang, W. Xu, Y. Zhu, and Y. Zhou, "Focal inverse distance transform maps for crowd localization," *CoRR*, vol. abs/2102.07925, 2021. [Online]. Available: https://arxiv.org/abs/2102.07925

[29] Microsoft, *Image APIs*, 2024, accessed: 2024-12-20. [Online]. Available: https://microsoft.github.io/AirSim/image_apis/

[30] D. Hendrycks and T. G. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *CoRR*, vol. abs/1807.01697, 2018. [Online]. Available: http://arxiv.org/abs/1807.01697

[31] M. B. Zafar, I. Valera, M. G. Rogriguez, and K. P. Gummadi, "Fairness Constraints: Mechanisms for Fair Classification," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. PMLR, 20–22 Apr 2017, pp. 962–970. [Online]. Available: https://proceedings.mlr.press/v54/zafar17a.html

[32] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 22–31. [Online]. Available: https://proceedings.mlr.press/v70/achiam17a.html