Journal of Aeronautics and Space Technologies

18(2) (2025) 76-93

Journal Homepage: jast.hho.msu.edu.tr

RESEARCH ARTICLE

Automated Landing Aid System Based on Visual Detection

Görsel Tespite Dayalı Otomatik İniş Yardım Sistemi

Ezgi Topçu[1], Emin Kaan Güreşçi[1], Ertuğrul Saatçı[1*]

[1] *Istanbul Kültür University, Department of Electrical and Electronics Engineering, 34158 Bakirkoy, Istanbul, Türkiye*

Received: *August 9, 2024* ***Revised:*** *July 2, 2025* ***Accepted:*** *July 9, 2025*

Abstract

Since the invention of the aircraft, the need for runways for takeoff and landing only overland has limited their uses. However, this situation has changed with the invention of aircraft carriers. An aircraft carrier with its flight deck is a warship that acts as an open-sea air base, and it is used to conduct operations in the designated areas. In this work, a prototype landing assistance system is proposed to work alongside landing signal officers (LSOs) on aircraft carriers. The system is based on visual information from the approach and landing phases and includes a voice assistant. The main goal of this system is to assist the LSOs rather than replace them by reducing their workload and minimizing fatigue-related human errors. The system is designed to be activated or deactivated by the LSO as required, ensuring flexibility and adaptability to varying operational needs. The proposed system is designed to be easy to integrate with the equipment already on aircraft carriers while keeping mobility in mind and tested in different weather and lighting conditions in a simulation environment.

Keywords: Aircraft Landing Systems, Landing Signal Officers, Machine Learning

Öz

Hava araçlarının icadının ardından kalkış ve iniş için ihtiyaç duyulan pistlerin sadece karada konumlandırılması kullanım alanlarını sınırlamıştır. Ancak bu durum uçuş güvertesi ile açık deniz hava üssü görevi gören ve belirlenen alanlarda operasyonlar yürütmek üzere kullanılabilen savaş gemilerinin icadı ile değişmiştir. Bu çalışmada, uçak gemilerinde iniş sinyal subayları (LSO) ile birlikte çalışmak üzere tasarlanmış bir iniş destek sistemi prototipi önerilmektedir. Prototip sistem, yaklaşma ve iniş aşamalarında elde edilen görsel bilgilere dayanmakta olup bir sesli uyarı asistanı içermektedir. Önerilen sistemde temel amaç, LSO'ların yerini almak yerine onların iş yüklerini azaltarak onlara yardımcı olmak ve yorgunlukla ilgili insan hatalarını en aza indirmektir. Değişen operasyonel ihtiyaçlara esneklik ve uyarlanabilirlik sağlamak amacıyla sistem ihtiyaç halinde sistem LSO tarafından etkinleştirilebilecek veya devre dışı bırakılacak şekilde tasarlanmıştır. Önerilen sistem, taşınabilirlik göz önünde bulundurularak uçak gemilerinde halihazırda bulunan ekipmanlarla birlikte kolayca kullanılabilecek şekilde tasarlanmış olup bir simülasyon ortamında farklı hava ve aydınlanma koşullarında test edilmiştir.

Anahtar Kelimeler: Uçak İniş Sistemleri, İniş Sinyal Görevlileri, Makine Öğrenmesi

*Corresponding Author
E-mail: e.saatci@iku.edu.tr

1. INTRODUCTION

Modern naval operations require the use of aircraft carriers, which offer a transportable platform for launching and retrieving aircraft throughout a range of mission types. Even seasoned pilots need the assistance of cutting-edge landing aid technologies to execute safe and accurate landings on the flight deck of a carrier, which is a difficult and demanding assignment for pilots. In the past, optical landing systems (OLS) [1], mirror landing systems (MLS) [2], deck edge lighting systems (DELS) [3], and precision approach radar (PAR) [4] have been the primary types of landing aid systems used to give pilots visual signals and guidance throughout the approach and landing phases.

A major disadvantage of these systems is their sensitivity to negative weather conditions. Negative weather conditions, such as fog, rain or low visibility can significantly affect the accuracy and effectiveness of these landing systems. Also, these weather conditions can limit the visibility of visual cues, interrupt radar signals or damage the equipment used in these systems, making it difficult for pilots to land the aircraft safely.

Current developments in computer vision and machine learning (ML) technology have created new opportunities for enhancing carrier landing aid systems by reducing the workload of the landing signal officer and the possible human errors. Additionally, ML-based landing aid systems have a potential to give pilots real-time instructions based on visual information from the approach and landing phases. This can enhance landing precision and safety under a variety of circumstances.

The landing process on aircraft carriers is named as recovery operations in USA navy. Recovery operations are predicated on case operations. There are 3 types of recovery in landing processes. Case I recoveries are practicable in weather conditions with a minimum cloud ceiling of 3000 feet and more than 5 nautical miles (nm) of visibility [5]. Case II recoveries are performed in daylight hours when conditions are worse than Case I limitations but adequate for performing visual landings. The cloud ceiling must be higher than 1000 feet and the visibility greater than 5nm [6]. Case III recovery is performed under a cloud ceiling of 1000 feet and/or visibility of less than 5nm. Regardless of weather conditions, Case III recovery can only be performed during nighttime operations [7].

Yue Meng et al. proposed a visual/inertial integrated algorithm for carrier landing based on POSIT and inertial navigation systems [8]. This approach focuses on integrating sensor fusion for landing precision but does not explore end-to-end ML-based detection or automation. Zhou et al. designed a visual pose tracking system using monocular vision for airborne targets [9]. However, this system is not tailored for carrier landing and does not involve autonomous decision-making. A few studies explore simulation frameworks and control algorithms for automatic landing. For instance, reinforcement learning approaches for longitudinal landing control have been explored in simulation [10]. To our knowledge, there is no study in the literature about automated landing aid systems used on aircraft carriers that are proposed to work alongside landing signal officers and that include a voice assistant.

In this work, we propose a prototype automated landing assistance system by utilizing ML supplied with visual references to enhance landing accuracy and safety. Specifically, we explore how ML algorithms can be trained on visual data from the landing aid system to provide pilots with real-time voice assisted guidance during the approach and landing phases. To achieve this, we process visual data from the camera located on the aircraft carrier's runway. The performance of the proposed system was evaluated with test scenarios in the Digital Combat Simulator test environment, and the evaluation was done according to the furthest distance that the prototype can recognize the aircraft under different weather and lighting conditions.

2. METHODOLOGY

2.1. Autonomous Aircraft Object Detection

Autonomous object detection enables machines to recognize and categorize items in their surroundings autonomously, without human assistance. Several products, including self-driving cars, drones, robotics, and surveillance systems, utilize this technology. In this study, an autonomous object detection algorithm uses live video provided by the camera to identify the position of the aircraft with respect to the center of the video footage in terms of horizontal and vertical distances in pixels. After determining the center point of the bounding box surrounding the aircraft on the video frame, the rest involves calculating the approximate distance between the aircraft and the aircraft carrier. With this information, the proposed system will determine appropriate radio callouts to warn the pilot as desired. Some examples of the voice commands given by the system include "power", "you're high", "keep going", "you're lined up" etc. Detailed information about radio callouts used in the system is given in Section 2.4.

In the literature, it is widely accepted that the progress of object detection has generally gone through two historical periods: the traditional object detection period (before 2014) and the deep learning-based detection period (after 2014) [11]. Most of the early object detection algorithms were built based on handcrafted features. Due to the lack of effective image representation at that time, people had to design sophisticated feature representations and various speedup skills. In recent years, the rapid development of deep learning techniques has greatly promoted the progress of object detection, leading to remarkable breakthroughs [11]. For up-to-date algorithms and their comparison, we refer the readers to [11].

Generally, there are two main approaches to detecting an aircraft automatically: traditional machine learning algorithms and deep learning-based object detection algorithms. Deep learning algorithms, such as Faster R-CNN [12], SSD [13] and YOLO [14] outperform traditional machine learning algorithms in terms of speed and accuracy [15]. To choose one object recognition algorithm over another, there are some important performance-related criteria, like frame rate per second (FPS) value, accuracy rate of the algorithm, and real-time object tracking capability. The Regions with CNN (R-CNN) algorithm uses selective search to determine object-containing regions and employs CNN classifiers to classify and locate objects. It can identify around 2000 objects but has a lower FPS rate due to its complexity. The Faster R-CNN algorithm, an improved version, still falls short of providing a sufficient FPS rate for real-time object recognition. The SSD algorithm has a higher FPS rate than Faster R-CNN, but it has a lower accuracy rate of

74.3% in object detection compared to the Faster R-CNN algorithm's 76.4% accuracy rate on the PASCAL VOC 2007 test set [16].

Our aim is to design a system that can be easily integrated with the equipment already on aircraft carriers while keeping mobility in mind. Lightweight network use is one of the methods to speed up an object detector and also reduce computation needs. In contrast to Faster R-CNN, YOLO is a one-stage detector that can retrieve all objects in one-step inference. One-stage detectors are well-liked by mobile devices with real-time and easy-to-deploy features. Therefore, YOLO is the method of choice for the required object detection. Additionally, the YOLO algorithm is superior to the other two in terms of the criteria mentioned, which led to its selection. However, YOLO may struggle to recognize small objects in early versions like v2 and v3 compared to its competitors [15]. This disadvantage can be mitigated by increasing the number of samples in the dataset and utilizing data augmentation techniques. But difficulties with small objects have been solved in later versions of YOLO. Table 1 provides a comparative list of the advantages and disadvantages of YOLO, Faster R-CNN, and SSD algorithms. Here, it is important to bear in mind that technology is constantly evolving; therefore, any comparison can become obsolete quickly. Every year, either new algorithms or updates to existing ones are published.

Table 1. Advantages and disadvantages of the three algorithms.

Algorithm	Advantage	Disadvantage
Faster R-CNN	The Region Proposal Networks (RPNs) that share convolutional layers with the object detection networks allow object detection to be almost real-time.	Despite the efficiency of the algorithm, it is not fast enough to be used in applications that require real time.
SSD	The use of a single network makes detection of the location of the objects faster than the Faster R-CNN method.	The detection accuracy of the objects is lower compared to the Faster R-CNN method.
YOLO	Detection of the location of objects is computationally efficient, allowing its use in real-time applications	The method has difficulties correctly detecting small objects in YOLOv2 and YOLOv3 [15], but not in later versions.

The YOLO algorithm is divided into versions within itself. The work in [17] systematically compares various lightweight YOLO versions, from YOLOv3 to YOLOv10, combined with two real-time trackers (DeepSORT and StrongSORT) on two challenging real-time datasets (MOT17 and MOT20). Authors conclude that YOLOv5 demonstrated robust proficiency in tracking, especially for small objects in congested environments, while ensuring high ID switch consistency. Additionally, they say that the performance of YOLOv5 with DeepSORT and StrongSORT is better than newer YOLO versions because it provides detections that are faster, more stable, and more consistent, all of which are essential for tracking. Moreover, the newer YOLO versions (YOLOv6–10) focus more on detection accuracy with anchor-free methods and transformers; however, these modifications have the potential to produce slight fluctuations in detection, which can result in less reliable tracking [17].

In light of the studies discussed above, the YOLOv5s model, which is also the fastest and smallest of the most up-to-date versions, was selected. YOLOV5s is a lightweight type of YOLOv5 algorithm. YOLOv5s used with a Military Aircraft Detection Dataset [18] for this study, outperformed the other algorithms (YOLOv5m, YOLOv5l and YOLOv5x).

2.2. Centroid Tracker

For real-time tracking of aircraft, the "Centroid Tracking" method [19] is used simultaneously with the YOLOv5s model. Specifically, the choice of centroid tracking was guided by its computational efficiency and ease of integration into a lightweight, real-time system suitable for embedded or resource-limited environments. While DeepSORT and StrongSORT offer higher tracking accuracy, they introduce additional computational overhead due to their reliance on appearance features and Kalman filtering. Given that our system already achieves sufficient tracking performance with YOLOv5s in relatively uncluttered environments (i.e., approach phases of aircraft with low object density), the simpler centroid tracker was deemed sufficient and more aligned with our design objectives of mobility and real-time responsiveness.

The centroid tracking method assigns identification numbers (IDs) to detected objects, and it is available in the OpenCV library. The steps of the centroid tracking algorithm can be summarized as below:
Step 1: Calculate the center of detected objects using the surrounding bounding box coordinates.
Step 2: For each subsequent frame, calculate the center of the detected objects using the bounding box coordinates and assign an ID to each newly detected object.
Step 3: Assume that the object with the minimum travel distance is the same object. In subsequent frames, identify the center point pairs with the smallest distance as the same object.
Step 4: Assign the same IDs to the moved center points to indicate that they represent the same object.

2.3. Decision Making

In the decision-making part, the outputs of the object detection algorithm and IDs assigned by the centroid tracker are used to determine the pilot's current position and then to guide the approaching pilot. The decision-making part uses the following parameters as inputs:

- Center point of the aircraft in terms of both vertical and horizontal distances in pixels relative to the top left corner of the camera image,
- Width and height of the bounding box in pixels,
- Distance of the aircraft to the camera in meters,
- Distance in pixels between the center of the aircraft bounding box and the center of the camera image,
- Assigned ID to every aircraft that exists in the camera image.

Then, the decision-making part gives the following output parameters:

- ID of the aircraft closest to the center of the camera image,
- Distance in meters between the camera and selected aircraft,
- Distance in pixels between the center of the camera image and the center of the selected aircraft bounding box.

Then, the decision-making part uses the output parameters to determine the appropriate radio calls to help the pilot correct errors in the aircraft's glide slope. Figure 1 shows the flowchart of the algorithm used in the proposed system.

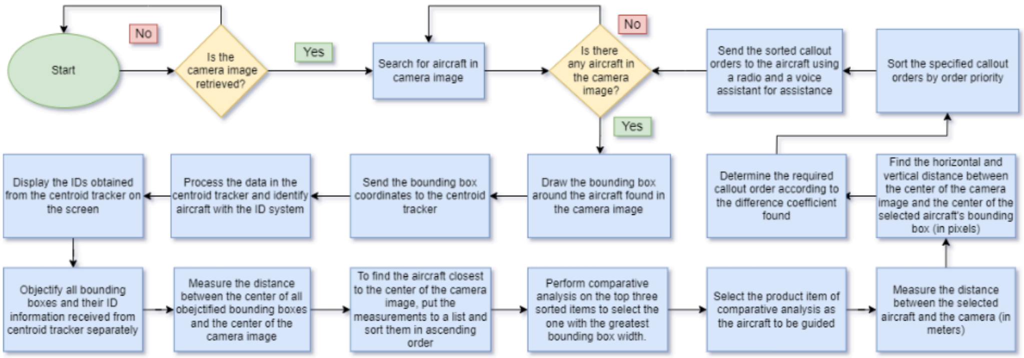

Figure 1. Flowchart of the proposed algorithm.

2.3.1. Selection of the Aircraft to be Guided Through Landing Process

The algorithm used for selecting an aircraft to guide during the landing process calculates the distance between the center of the camera image and the center of each aircraft in the image. Then, the algorithm arranges these distance values in ascending order to create a list. The aircraft with the smallest distance from the center of the image is chosen as the one to be guided by the algorithm. The steps of the algorithm can be summarized as below:

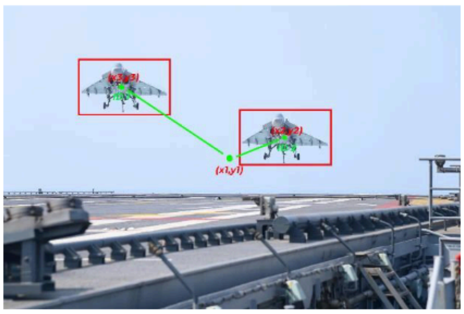

Figure 2. Euclidean distance between the center of the camera image and the center of the bounding boxes surrounding aircraft.

Step 1: Use (1) to calculate the Euclidean distance between the center of the camera image and the center of the bounding boxes surrounding the aircraft, as shown in Figure 2.

$$|(x_1, y_1), (x_2, y_2)| = \sqrt{((x_2 - x_1)^2 + (y_2 - y_1)^2)} \tag{1}$$

In Figure 2, (x_1, y_1) corresponds to the center of the camera image, and (x_2, y_2) and (x_3, y_3) correspond to the centers of the bounding boxes surrounding the two aircraft that are visible in the figure.

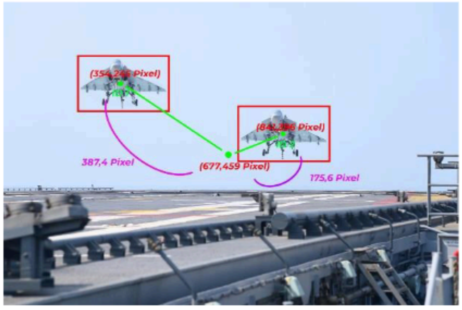

Figure 3. An example camera image showing the distance values between the center of the image and the center of the bounding boxes.

Step 2: Store the measured distance values for all aircrafts that are visible in the camera image with their unique ID number in a list, then sort distances in ascending order.

Figure 3 shows an example camera image with a size of 1366 by 908 pixels. In the figure, the point with coordinates (677,459) corresponds to the center point (x_1, y_1), and the points (354,245) and (841,396) are the center point coordinates (x_2, y_2) and (x_3, y_3) of the two aircraft that are visible in the figure. The Euclidean distance between (x_1, y_1) and (x_2, y_2) is measured as 175.6 pixels, and the distance between (x_1, y_1) and (x_3, y_3) is measured as 387.4 pixels.

Step 3: From the sorted list, identify the top three aircraft positioned closest to the center of the image based on their measured distances. Conduct a comparative analysis among these candidates by examining the width of their respective bounding boxes. The aircraft exhibiting the greatest bounding box width is subsequently designated as the target to be guided by the algorithm. Following this selection, the algorithm retrieves and returns the corresponding ID along with all other relevant parameters associated with the selected aircraft. According to the example given in Figure 3, the aircraft ultimately selected has a distance of 175.6 pixels from the center of the image, and its assigned ID is 2.

2.3.2. Measuring the Distance Between Camera and Aircraft

This section explains how to measure the distance between the camera and the aircraft using the "Size-Distance Scaling Equation" [20] method. The size-distance scaling equation given in (2) is used to calculate the distance D between the camera and the aircraft. This method relies on several key factors, such as the wing width P of the aircraft in meters, the distance D between the camera and the aircraft in meters, the object's size

R on the camera sensor in millimeters, and the focal length k of the camera lens in millimeters.

The prototype system is proposed to work alongside LSOs on aircraft carriers; therefore, the aircraft's type (and therefore wing width P) is entered into the algorithm by the LSO from the list of predetermined aircraft types that are included in the prototype system. This method prevents the proposed system from identifying the incorrect type of aircraft by removing the requirement for the object detection algorithm to discriminate between different aircraft types correctly.

Using the size R of the object's image on the camera sensor, the distance D between the camera and the object can be calculated with one camera by using (2).

$$\frac{P}{D} = \frac{R}{k} \tag{2}$$

A visualization of (2) is shown in Figure 4.

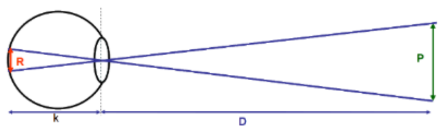

Figure 4. Visualization of the size-distance scaling equation.

First, the object size R_p is calculated in pixels and then converted to the R value in millimeters using (3).

$$R = R_p * p_x \tag{3}$$

In (3), the size p_x of the pixel on the camera sensor is calculated by (4), where L is the sensor horizontal length in millimeters and N_x is the number of pixels along the horizontal axis.

$$p_x = \frac{L}{N_x} \tag{4}$$

The distance measurement method depends on the accurate alignment of the aircraft in the image and the surrounding bounding box. If the size of the bounding box is larger than what it is supposed to be, for example, if the aircraft to be detected has a width of 90 pixels but the bounding box covering the aircraft has a width of 100 pixels, then the distance of the aircraft from the camera will be measured as 198 meters instead of 220 meters. At a distance of 220 meters, the center of the aircraft's bounding box, which is assumed to be 2 degrees below the center point of the image, will be located 6.91 meters from the center of the image. But at the measured distance of 198 meters, the same centerline distance will be calculated as 7.67 meters. Since both distances are within the margin of error allowed for the pilot (10 meters), no pilot warning is required.

2.3.3. Guidance of the Aircraft

This section explains how the proposed system determines the appropriate callouts based on the aircraft's current location. Basically, the algorithm aims to place the pilot on the glide slope, i.e., closer to the camera center, using the LSO's callouts given in Table 2.

Table 2. LSO's callouts and their meanings in the proposed system.

Callouts	Meanings
You're high	Aircraft is far above the glidepath.
Power	Aircraft is far below the glidepath.
Come left	Aircraft is right of the centerline.
Come right	Aircraft is left of the centerline.
You're in line	Aircraft is on the centerline.

The algorithm is based on the ratio of camera pixel count in terms of width and height to the camera's angle of view in degrees. With the help of the distance between the aircraft and the camera and the angle information, the algorithm tries to calculate the distance between the aircraft and the glide slope position where the aircraft should be. The steps of the algorithm are listed below:

Step 1: Receive the aircraft's location in the camera image relative to the center of the image via object tracking.

Step 2: Calculate horizontal and vertical distances to determine how far the aircraft is from the center of the image in pixels and also determine the aircraft's position (lower, higher, left, or right) based on the sign of these distances.

Step 3: Using camera resolution, angle of view, and the distance between the camera image center and the aircraft center, calculate the angles between the aircraft center and the camera center for both the horizontal and vertical axes with the help of (5), where θ_{c_axis} is the angle relative to the center of the camera image in degrees, d_{cc} is the distance between the center of the camera and the center of the aircraft in one axis in pixels, α_{axis} is the camera angle of view of the axis in degrees, and r_{axis} is the camera resolution along the axis in pixels.

$$\theta_{c\_axis} = \frac{|d_{cc}| * \alpha_{axis}}{r_{axis}} \tag{5}$$

Step 4: Use the angles and the distance between the camera and the aircraft and form triangles to calculate the real horizontal and vertical distances in meters between the aircraft and the optimal point on the glide slope, as shown in Figure 5.

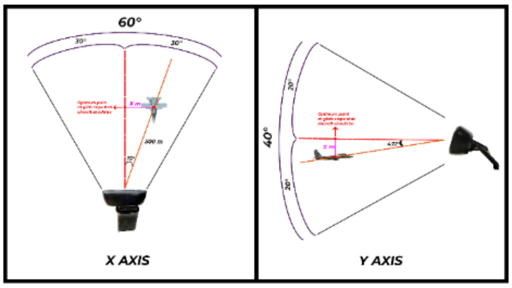

Figure 5. The triangles used to calculate horizontal and vertical distances between the aircraft and the optimum point on glide slope.

Step 5: After calculating the real distances for both axes, compare them with the user-defined maximum allowable error. If the error exceeds the user-defined limit on any axis, select an appropriate callout to correct the aircraft's position.

2.4. Voice Assistant

This section describes how, and in what order, the selected callouts should be transmitted to the approaching aircraft. As pilot safety is the primary objective, it is extremely important that the orders be transmitted as sequentially as possible. Therefore, the callouts selected for transmission are transmitted in an ordered sequence from the highest priority callout to the lowest priority callout. Callout priorities are listed in Table 3.

Table 3. Callout priorities.

Callout Priority	Callout
1	Power
2	Come left
3	You’re high
4	Come right
5	You’re in line

3. IMPLEMENTATION AND SIMULATION RESULTS

3.1. Integration of the System into the Simulation Environment

To evaluate the performance of the proposed system under realistic operational conditions, a comparative analysis was conducted between two widely used simulation platforms: Microsoft Flight Simulator 2020 (MSFS 2020) and Digital Combat Simulator (DCS). The comparison was based on the following core technical benchmarks:

- Precision and customizability of weather conditions
- Flexibility in configuring camera perspectives
- Fidelity of the physics engine to real-world dynamics
- Availability and realism of military-grade aircraft models

Both simulation environments support the modeling of fighter aircraft (e.g., F/A-18) and allow for the configuration of basic weather parameters. However, when evaluated in terms of environmental control granularity, physics accuracy, and visual perception fidelity, DCS demonstrates clear advantages over MSFS 2020.

First, DCS provides fine-grained control over atmospheric conditions, such as wind speed across altitude layers, turbulence, cloud density, and visibility—offering an ideal framework for controlled experimentation. In contrast, MSFS 2020 primarily relies on real-time server-based weather data, which can limit repeatability and test consistency.

Second, DCS supports aircraft carrier operations with fully functional landing systems, including arresting gear, catapult mechanisms, and dedicated LSO (Landing Signal Officer) interfaces—essential elements for testing aircraft approach behavior. MSFS 2020 lacks native support for carrier landings, relying instead on third-party modifications that are often limited in fidelity.

Lastly, the physics engine of DCS—particularly its Professional Flight Model (PFM)—offers superior realism, enabling high-fidelity replication of aircraft dynamics, even under edge-of-envelope conditions. MSFS's flight model, while suitable for civilian simulation, emphasizes accessibility and smooth control, which can diminish its applicability in high-performance or tactical use cases.

Based on these criteria, Digital Combat Simulator was selected as the simulation platform for this study due to its robustness, configurability, and higher degree of realism. The integration of the proposed system with the DCS platform was achieved using the screen capture functionality of the OpenCV library in Python. This module was employed to capture and transmit, in real time, the visual feed from the simulation environment—specifically, the LSO's camera view during carrier landing operations. The proposed system then processed this visual input and generated real-time guidance instructions for the pilot based on the aircraft's position and dynamics.

To provide further clarity on how the system perceives approaching aircraft within the simulated environment, illustrative samples from tests conducted under various weather conditions are presented under Section 3.2 in Figures 6-11.

3.2. System Tests Under Different Weather Conditions

To measure the performance of the proposed system, tests were carried out in different weather conditions. To measure the performance, the criterion is the maximum distance at which the proposed system can recognize the approaching aircraft. The reason is that if the proposed system starts to guide an aircraft at the farthest visible distance, it will have extra time to send more guidance commands than at shorter distances. This extra time allows more chance to correct maneuvering errors that can be caused by pilot error.

During the tests of the proposed system, the confidence threshold of the object detection algorithm was determined to be 0.8 with the help of the confidence interval formula. Our test criterion is the distance in meters at which the aircraft guidance system recognizes the aircraft.

The proposed system was tested with different airplane types in different weather conditions. The maximum distance in meters from which guidance starts is listed in Table 4 under different weather and lighting conditions.

Table 4. Maximum guidance distance for different aircraft types at different weather conditions.

Aircraft Type / Weather Conditions	F-18	F-35	Su-33
Ideal Conditions	3017 m	2972 m	3400 m
Foggy	2946 m	2889 m	3275 m
Rainy	2980 m	2876 m	3252 m
The Sun in Front of the Camera	2308 m	2270 m	2544 m
The Sun Behind the Camera	2556 m	2469 m	2813 m
The Sun is on the Right of the Camera	2857 m	2786 m	3276 m
The Sun is on the Left of the Camera	2494 m	2357 m	2871 m
Sunrise	2347 m	2310 m	2765 m
Sunset	2666 m	2587 m	2678 m
Average distance	2685,6 m	2612,8 m	2986 m

As can be seen from Table 4, the proposed system recognizes the approaching aircraft at a distance longer than 1.2 nautical miles (2222.4 m), which is the minimum safe landing guidance distance.

Representative images from tests conducted under diverse weather conditions are presented in Figures 6-11 below, accompanied by corresponding measurements obtained within the Tacview flight data analysis tool.

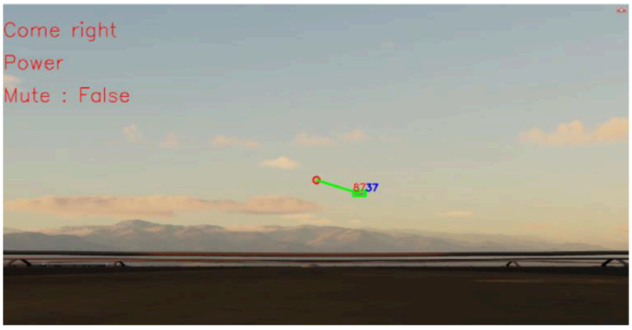

Figure 6. System maximum guidance distance test in sunrise conditions.

Figure 7. Tacview measurement of system maximum guidance distance in sunrise conditions.

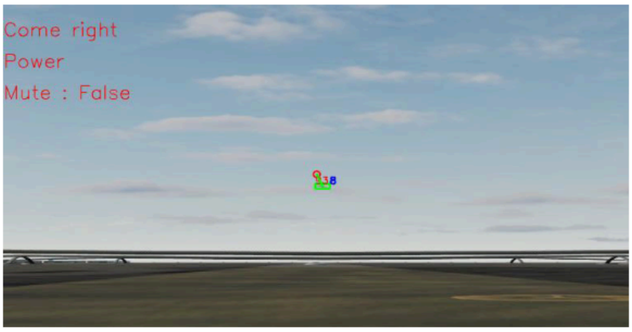

Figure 8. System maximum guidance distance test in sunset conditions.

Figure 9. Tacview measurement of system maximum guidance distance in sunset conditions.

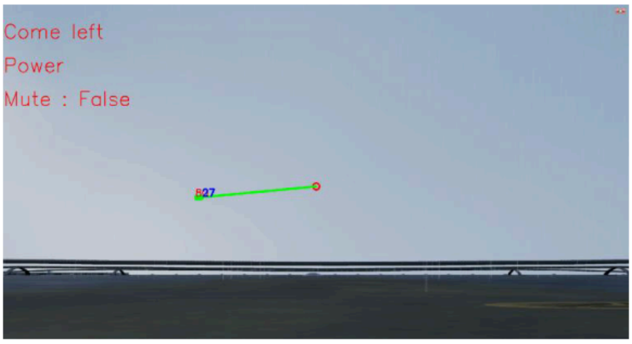

Figure 10. System maximum guidance distance test in rainy conditions.

Figure 11. Tacview measurement of system maximum guidance distance in rainy conditions.

3.3. System Measurement Accuracy Test

To determine the measurement performance of the proposed system, a test was carried out in the simulation environment by comparing the distance data from Tacview with the distance data obtained from the proposed algorithm. In the test, the performance criterion is the matching of the measurement data obtained from the proposed algorithm with the actual distance obtained from Tacview. Tacview is a universal flight data analysis tool used in briefing and debriefing for visualization purposes [21]. An example Tacview screen showing the measured distance between the camera of the proposed system and the aircraft in the simulation is shown in Figure 12.

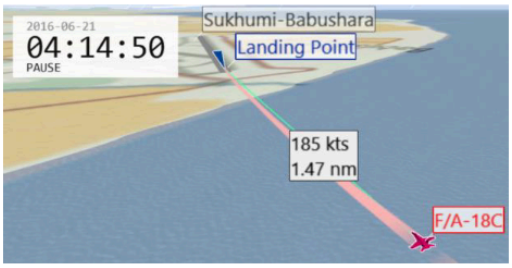

Figure 12. An example Tacview screen showing the measured distance between the camera of the proposed system and the aircraft in the simulation.

For this purpose, four waypoints with distances of 1.4, 1, 0.7, and 0.2 nautical miles from the camera of the proposed system were selected to compare the measurements. At these waypoints, the error rate was obtained by comparing the measurements made by the proposed system with the distances obtained from Tacview. The distance measurements made by the proposed system, the distances obtained from Tacview, and the error rates obtained by comparing these distances are listed in Table 5.

Table 5. Comparison of the distances from the camera obtained by the proposed system and the real distance given by Tacview.

Measurements / Waypoints	Measured Distance from the Camera	Real Distance from the Camera	Error Rate
Waypoint 1	2567 m	2592,8 m (1,4 nm)	%0,995
Waypoint 2	1833 m	1852 m (1 nm)	%1,025
Waypoint 3	1282 m	1296,4 m (0,7 nm)	%1,110
Waypoint 4	366 m	370,4 m (0.2 nm)	%1,187

As can be seen from Table 5, measured and real distances are almost the same, and the error rate is around 1%.

4. CONCLUSION

During the recovery operations, the stress level on the LSO may arise and lead to fatigue-related human errors. The aim of this work is to develop a prototype landing assistance system working alongside LSOs on aircraft carriers to increase the safety of flight by reducing the human errors caused by fatigue. For this purpose, we proposed a semi-supervised voice-assisted runway alignment system based on visual information from the approach and landing phases to lighten the workload of the LSO and let him/her better focus on the most important task, which is to abort the landing in dangerous situations. Therefore, the proposed system was developed to change the task structure of the traditional LSO and was designed to be easy to integrate with the equipment already on aircraft carriers. In order to lead the pilot to the correct landing path, we developed a single camera distance measurement and error correction algorithm based on the distance between the position of the approaching aircraft and the position it should be during the landing.

Experiments conducted in the simulation environment under different weather and lighting conditions have shown that the proposed system recognizes the approaching aircraft at a distance longer than 1.2 nautical miles (2222.4 m), which is the minimum safe landing guidance distance. Therefore, the proposed algorithm is a candidate to be used in CASE I (minimum visibility of 5 nautical miles) and CASE II (minimum visibility of 1-3 nautical miles) recovery operations in its current form. In the remaining CASE III landing type, the proposed algorithm is inoperable due to the lack of visual approach during the operations. The main reason for this is that the proposed algorithm can only work if it can establish a visual contact with the aircraft, and the visibility distance in CASE III type recovery operations is less than the minimum safe landing guidance distance.

As can be seen from the results, the rainy weather condition is the nearest condition to the performance of the ideal condition, while the condition where the sun is in front of the camera is the farthest condition to the performance of the ideal condition. Here, the factor that affects the performance is the amount of light falling on the camera's lens. Although the conditions for a safe landing are met, the low rates can be further increased by using different filtering methods. Additionally, tests conducted in the simulation environment to determine system measurement accuracy have shown that the measured and real distances are almost the same, and the error rate is around 1%.

In the future, we plan to use infrared detection and laser distance measurement sensors in the camera to manage to guide aircraft at longer distances. This would allow for better object detection in dark environments and increase the object detection distance to greater distances.

REFERENCES

[1] R. H. Shields, *The Depth of Flash Optical Landing System*, Defense Technical Information Center, Naval Research Lab Washington D C, Interim Report, Feb. 3, 1967. Accessed: August 26, 2023 [Online]. Available: https://apps.dtic.mil/sti/pdfs/AD0651886.pdf

[2] A. Spencer, “Landings are Mandatory: Lt. Don Engen and the Mirror Landing System,” National Air and Space Museum. Accessed: August 26, 2023 [Online]. Available: https://airandspace.si.edu/stories/editorial/don-engen-mirror-landing-system

[3] C. Best, *The Influence of Ship Deck-Edge Lighting on Perception of Position and Movement During Helicopter Recovery*, Defence Science and Technology Organisation Victoria (Australia), Platform Sciences Lab, Air Operations Division, Technical Report, Jul. 1, 2006. Accessed: August 26, 2023 [Online]. Available: https://apps.dtic.mil/sti/pdfs/ADA462950.pdf

[4] N. Chand, “Precision Approach Radar,” SP's Naval Forces. Accessed: August 26, 2023 [Online]. Available: https://www.spsnavalforces.com/story/?id=236

[5] K. H. Delano, “Air Traffic Control Doctrine - Case I”, in Naval Air Training Command, Flight Training Instruction, Shipboard Operations and Search and Rescue TH-57, Nas Corpus Christi, Texas, CNATRA P-429, United States Navy, pp. 15, 2021.

[6] K. H. Delano, “Air Traffic Control Doctrine - Case II”, in Naval Air Training Command, Flight Training Instruction, Shipboard Operations and Search and Rescue TH-57, Nas Corpus Christi, Texas, CNATRA P-429, United States Navy, pp. 15, 2021.

[7] K. H. Delano, “Air Traffic Control Doctrine - Case III”, in Naval Air Training Command, Flight Training Instruction, Shipboard Operations and Search and Rescue TH-57, Nas Corpus Christi, Texas, CNATRA P-429, United States Navy, pp. 16, 2021.

[8] Y. Meng, W. Wang, Z. Ding, “Research on the visual/inertial integrated carrier landing guidance algorithm,” *International Journal of Advanced Robotic Systems*, vol. 15, no. 2, pp. 1-9, 2018.

[9] J. Zhou, Q. Wang, Z. Zhang and X. Sun, “Aircraft Carrier Pose Tracking Based on Adaptive Region in Visual Landing,” *Drones*, vol. 6, no.7, pp. 1-17, 2022.

[10] J. Wang, H. Tang, H. Liu, L. Wang, D. Chen, Y. Lu, “Design of Longitudinal Automatic Carrier Landing System Based on Reinforcement Learning,” In L. Yan, H. Duan, Y. Deng (eds), Advances in Guidance, Navigation and Control ICGNC 2022, Lecture Notes in Electrical Engineering, vol. 845, Springer, Singapore, 2022.

[11] Z. Zou, K. Chen, Z. Shi, Y. Guo and J. Ye, "Object Detection in 20 Years: A Survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257-276, 2023.

[12] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real- Time Object Detection with Region Proposal Networks," *IEEE TPAMI*, vol. 39, no. 6, pp. 1137-1149, 2017.

[13] W. Liu, D. Anguelov, D. Erhan, C. Reed, S. Reed, C.-Y. Fu and A. C. Berg, "SSD: Single Shot Multibox Detector," presented at European Conference on Computer Vision – ECCV2016, Springer, pp. 21-37, Sep. 17, 2016.

[14] P. Jiang, D. Ergu, F. Liu, Y. Cai and B. Ma, "A Review of Yolo Algorithm Developments," presented at The 8th International Conference on Information Technology and Quantitative Management, Chengdu, China, 2022.

[15] R. Kaur, S. Singh, "A comprehensive review of object detection with deep learning," *Digit. Signal Process.*, vol. 132, pp. 1-17, 2023.

[16] S. A. Sanchez, H. J. Romero and A. D. Morales, "A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 844, pp. 1-15, 2020.

[17] K. Alkandary, A. S. Yildiz and H. Meng, "A Comparative Study of YOLO Series (v3–v10) with DeepSORT and StrongSORT: A Real-Time Tracking Performance Study," *Electronics*, vol. 14, no. 5, pp. 1-27, 2025.

[18] T. Nakamura, *Military Aircraft Detection Dataset*, Accessed: August 26, 2023 [Online]. Available: https://www.kaggle.com/datasets/a2015003713/militaryaircraftdetectiondataset.

[19] H. Zhou, J. He and Z. Huang, "Multiple Centroid-Based Multi-Object Tracking by Decision Making," Presented at International Conference on Computer Science, Electronics and Communication Engineering - CSECE 2018, China, pp. 74-78, 2018. Accessed: August 26, 2023 [Online]. Available: https://www.atlantis-press.com/article/25893359.pdf

[20] C. J. Erkelens, "Perception of Perpective Angles," *I-Perception*, vol. 6, no. 3, pp. 1-11, 2015.

[21] RAIA SOFTWARE INC., "Tacview," Accessed: April 12, 2024 [Online]. Available: https://www.tacview.net/product/about/en/

To Cite This Article: E. Topçu, E. K. Güreşçi, E. Saatçı, *Automated Landing Aid System Based on Visual Detection*, Journal of Aeronautics and Space Technologies **18**(2), 76-93 (2025).

VITAE

Ezgi Topçu received the B.E. Electrical & Electronics Engineering from Istanbul Kultur University, Istanbul, Türkiye in 2023. She is currently an Project Engineer at Lightweight Platforms Department of Baykar Technologies. Her major interests are aviation, image processing and artificial vision.

Emin Kaan Güreşçi received the B.E. Electrical & Electronics Engineering from Istanbul Kultur University, Istanbul, Türkiye in 2023. He is currently working as a Cadet Pilot at

Turkish Airlines and continues his training at Ayjet Anadolu Yildizlari Flight School. His main areas of interest are aviation, artificial intelligence and UAV technologies.

Ertuğrul Saatçı received the B.S. and M.S. degree from Istanbul University, Istanbul, Türkiye, in 1993 and 1996, respectively, and Ph.D. degree from London South Bank University in 2003. He is currently an Assistant Professor at Department of Electrical and Electronics Engineering, Istanbul Kultur University. His major research interests lie in the areas of signal and image processing, biomedical signal processing, biometrics, and cellular neural networks.