Proceedings Paper:

Tsapparellas, K., Jelev, N., Waters, J. et al. (2 more authors) (2023) Vision-based runway detection and landing for unmanned aerial vehicle enhanced autonomy. In: 2023 IEEE International Conference on Mechatronics and Automation (ICMA) Proceedings. 2023 IEEE International Conference on Mechatronics and Automation (ICMA), 06-09 Aug 2023, Harbin, Heilongjiang, China. Institute of Electrical and Electronics Engineers (IEEE) , pp. 239-246. ISBN 9798350320855

https://doi.org/10.1109/ICMA57826.2023.10215523

Vision-based Runway Detection and Landing for Unmanned Aerial Vehicle Enhanced Autonomy

Kyriacos Tsapparellas*, Nickolay Jelev**, Jonathon Waters**, Sabine Brunswicker*** and Lyudmila S Mihaylova*
*Department of Automatic Control and Systems Engineering, University of Sheffield, S1 3JD, UK
**Windracers, Distributed Avionics, Healthaid House, Malborough Hill, London, HA1 1UD, UK
***Research Center for Digital Innovation, Purdue University, USA
k.tsapparellas@sheffield.ac.uk, njelev@windracers.org, jonathon.waters@distributed-avionics.com,
sbrunswi@purdue.edu, l.s.mihaylova@sheffield.ac.uk

***Abstract*—Introducing autonomy is a task of paramount importance and is currently investigated in many areas, especially for autonomous cars and Unmanned Aerial Vehicles (UAVs). Most UAVs are still remotely human-controlled. A necessity is to implement on-board solutions, able to work in all weather conditions and at any time. Hence, on this topic, we give an overview of recent advances for vision-based landing of UAVs. A thorough classification of the main recently developed methods is introduced with a discussion of their advantages and disadvantages. The paper presents a new solution for autonomous UAV vision-based landing, focusing on runway detection using a hybrid approach combining multi-image matching, SIFT and object tracking. The results are evaluated and validated using simulated images sampled with the X-Plane 11 flight simulator and real-world videos collected during automated flights performed by the ULTRA vehicle, one of the biggest UAVs in the UK [1]. The statistical analysis from the validation of the proposed approach shows a high level of accuracy around 94.89% in clear weather conditions and real-time computational performance.**

***Keywords*—** Unmanned Aerial Vehicles (UAVs), Autonomous Landing, Runway Detection, Autonomy, X-Plane 11 flight simulator, Computer Vision, Vision-based landing.

I. Motivation

Unmanned Aerial Vehicles (UAVs) have received increased attention in research and practice given their potential to solve economic and social challenges like search and rescue, wildlife protection, humanitarian aid [2] or other challenging tasks [1], [3]. Examples of economic and social benefits of UAVs compared to manned vehicles are, the increase in efficiency and flexibility of operations, labour cost reduction, and potentially greater safety. To realise such benefits, UAVs need to operate at an increased level of independence from a remote operator, even if they need to fly at an extended and beyond visual line of sight (EVLOS or BVLOS) [4]. Existing research and industry practice aims to tackle this problem using automation: Today, some UAVs do not require the active involvement of a remote pilot who flies and controls the UAV using remote operations software. Instead, they only require some degree of human monitoring as they can fly in an automated way with autopilots using classical airspace signals (typically from a radar, e.g. from the Automatic Dependent Surveillance–Broadcast (ADS-B) or the Global Navigation Satellite System (GNSS)) and human rules defined. However, such automated systems are not yet autonomous because they are unable to engage in more intelligent and self-directed actions in response to unpredictable events such as the unexpected appearance of another vehicle or the loss of radar signals. This lack of autonomous response to uncertainty creates safety concerns about using existing automated solutions in real-world settings [4], [5]. Thus, existing UAV research has yet to develop systems that provide an enhanced degree of autonomous intelligence and resilience to events that are unknown at the time of the system design [5]–[7]. Such events occur more often during real-world operations rather than controlled experimental setups in labs, widely used in today's UAV research.

Fig. 1: ULTRA [1], one of the biggest UAVs in the UK, capable of carrying 100 kilograms of payload

The level of autonomy of a UAV can be broadly defined by its ability to (1) operate at a high level of independence from human actions, (2) perform complex missions, and (3) respond to environmental uncertainty that characterises operations in real-life outside of university labs [5]. The need for a high level of autonomy is particularly important in response to unpredictable events that may cause a collision of a UAV with other vehicles (UAVs and manned vehicles) or even humans prior to or during landing as such events create significant safety concerns [7]. Thus, this paper is tackling this practical need and is proposing and validating a new vision-based solution for runway detection. That allows for landing in scenarios where existing automation fails to respond to unpredictable events during real-world operations. Motivated by the need to realise a high level of autonomy for a real-time automated solution.

This research presents results from the case of a large UAV already operating in an automated way at BVLOS using a rule-based autopilot. The ULTRA UAV (shown in Figure 1) designed by Windracers [1] for aid delivery is already operating in real-life conditions that encounters environmental uncertainty requiring complex mission operations. Using simulated as well as real-world flight data collected for this and with this vehicle, this research work proposes and validates a new method for vision-guided autonomous UAV navigation that outperforms methods proposed in applied UAV research or used in real-world applications [8].

The proposed approach for the UAV landing identifies the target runway using multiple reference images processes camera frames in real-time and provides a decision for proceeding with the landing or declining it. The analysis is performed in the image plane and the proposed system is developed for the purpose of landing monitoring without using a GNSS.

A. Contributions

The main contributions of this paper are the following: 1) Recent advances in the area of UAV vision-based navigation are discussed, 2) A new practical Computer Vision approach is proposed which is able to achieve real-time performance. The vision-based system works with optical video data, includes the scale-invariant feature transform (SIFT) [9] and recognises remotely the runway during the landing approach. 3) A thorough validation of the approach is performed over simulated and real video data. The real video data are from flights of the ULRA UAV [1] which is one of the biggest fixed-wing UAV built so far in the UK.

The rest of this paper is organised as follows. Section II refers to the related work, that exists on Computer Vision and deep learning methods for identifying a runway from a distance and assisting tools for the landing of a UAV. Moreover, a discussion of the pros and cons of deep learning methods and traditional vision methods is presented. Section III, discusses the potential of Computer Vision methods of providing a solution for identifying a runway during the landing approach of the UAV, including feature-based methods for object detection, classification and object tracking. Section IV present the proposed architecture for runway identification and the experiments that were conducted using simulation tools alongside the results obtained. Detailed evaluation and validation of the proposed architecture are given in Section V. Section VI, summarises the main results from the proposed system.

II. Related Work

A. Runway Detection Systems

Previous related work [10], [11] assumes that GNSS is currently installed on the majority of manned aircraft as well as UAVs. GNSS precision is dependent on local electromagnetic conditions, terrain and other temporal factors such as satellite availability. This paper is proposing an algorithm for runway detection using Computer Vision methodologies to enable error identification on the positioning of the UAV with respect to the runway and abort the landing. The paper explores the different conditions that the aircraft needs to land, such as low visibility, good, rainy weather and extreme weather such as wind.

Being able to detect the runway efficiently and precisely is considered a key step of automatic landing, which can trigger a warning to the UAV operator, trigger an automatic mission abort, or trigger an autonomous landing correction. Therefore, developing a suitable runway detection algorithm is crucial. In Akbar's paper [12], the author categorises approaches, template matching, Hough transform [13], Active Contours, and Machine Learning algorithms, into two main categories: template based and feature-based.

A real-time sensor-guided runway detecting system was put out by Wang [14]. In the beginning, a search region was established and a runway template was created using topographical data and sensor data from the "Synthetic Vision System" (SVS) and "Enhanced Vision System" (EVS). Following on from the query image, the original search region was used to apply the lines extraction approach. The final step involved matching the template and query images in order to identify the precise runway area.

An approach to quick detection of the airport runway in remote sensing photos is proposed by Yang [15]. The runway is extracted using the Otsu method [16], fractional differential gradient operator, and "Hough transform" (HT) [13]. The proposed method's ability to operate quickly and produce positive test results for significant increases in computing speed and decreased data operations were both demonstrated.

Autonomous landing for an unmanned aircraft platform depends on the quick and precise identification of its landing runway. Using an airborne camera to capture photos of the landing runway, Nazir et al. [17] used edge detection algorithms to determine the runway's precise location. The evaluation method proposed is based on the classification and identification of the runway as a class and the average processing time of the system.

B. Comparison Between Traditional Computer Vision and Deep Learning

In recent years, there has been a significant shift in the field towards the use of deep learning methods. In [18], a Faster R-CNN [19] approach was used for airport region detection. A convolutional neural network (CNN) is used to identify prospective airports, and a second CNN is detecting airports based on improved runway features. Line-segment detector (LSD) [20] is used for potential airport regions as described in [21]. The model that was used for classification over the regions is AlexNet [22].

A two-stage system was developed by Akbar et al. [12], that is extracting features on images using CNN and performs classification using a softmax classifier, on classes such as roads, forests, and runways. The next step of the approach is using Hough transform [13], line segmentation, to perform runway segmentation, for localisation purposes.

Traditional Computer Vision methods rely on hand-crafted features and rule-based systems to analyse images and videos. These methods often require a significant amount of domain knowledge and can be prone to errors in certain scenarios [23]. For example, traditional Computer Vision algorithms designed to recognise objects in images can fail when presented with images taken from different angles or under different lighting conditions [24]. Additionally, deep learning methods [23] open new avenues through their abilities to process big data and learn from them. Deep learning-based Computer Vision learn features and patterns from the data [24]. These models are able to handle a wide range of images and videos and can often achieve higher accuracy than traditional methods. For example, state of art performance was achieved from deep learning-based object detection algorithms on benchmark data sets such as COCO [25] and PASCAL VOC [26]. Additionally, deep learning methods will continue improving as they have access to more data, making them more adaptable to new scenarios [27], [28].

One major disadvantage of deep learning-based Computer Vision is the large amounts of labelled data and computational resources needed for training [29]. This a difficulty researchers and practitioners face when developing and deploying deep learning-based Computer Vision systems, especially in scenarios where there are limited data or computational resources [28]. Additionally, deep learning models can be difficult to interpret and understand, making it challenging to understand why a particular model is making a certain prediction [24].

In conclusion, both methodologies enclose advantages and disadvantages. Traditional methods may be more interpretable and require less data and computational power, but errors can be pruned and may not be able to handle a wide range of images and videos. On the other hand, deep learning-based Computer Vision can achieve higher accuracy and can continue to improve as they are exposed to more data, but they require large amounts of labelled data and computational resources to train. The choice of which approach to use will depend on the specific scenario and the available resources.

III. Methods

A. Data Collection Approaches

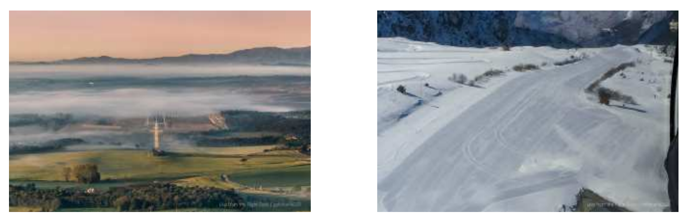
For the purpose of data collection for the task of runway detection, the available data sets are based on aerial or satellite images to train a network [12]. The ground truth of the runways is manually annotated in such data sets. However, Krajacic T. [30], used X-Plane 11 simulator as a visual system for research. The simulator can generate numerous scenarios of a flight such as different weather conditions. Day, night, strong winds, fog, rain, and ideal conditions can be simulated using X-Plane 11, which are realistic scenarios that a pilot and a plane can face during a flight. Bittar et al. [31] proposed the software in the loop software using the X-Plane and Simulink.

B. Feature-Based Approaches

Feature-based approaches are not based on a specific model that is detecting and tracks features like corners, edges, and other easily localised features [32] as shown in Figures 2a and 2b. Compared to template-based approaches, this approach can reduce the cost of creating a model. This approach might still be able to detect and localise runways, even if the poor weather condition blocks some of the runway features [10]. Especially, in poor weather conditions, snow and fog might block some necessary markings that may result in detection incorrectly, as shown in Figure 3a and 3b. There are feature-based approaches that have proven their advantages, such as the Scale Invariant Feature Transform (SIFT) [32], the Speeded-Up Robust Features Transform (SURF) [33] and edge detection approaches [34].

(a) (b)

Fig. 2: Runways with edges lines, centerline, landing designator and threshold



(a) (b)

Fig. 3: Runways during fog and snow weather conditions

Lowe's paper [32] proposed a machine learning algorithm named SIFT to match the object even if rotation, distortion, the addition of noise, and change in illumination. According to Lowe [32], SIFT can process images with near real-time performance, Hu [35] and Zhang [36] use it to match objects and register images, because of its robustness and performance. However, Daixian [9] analyses that the computational time and matching accuracy of the original SIFT are not good enough, therefore, an improved SIFT was proposed, and the enhancement of Real-time quality and stability of the algorithm is verified. As for the research on landing on a runway, Miller [37] used SIFT to detect the terrain as visual information, which can steer the UAV to the runway, before the runway is visible.

The main steps that are used for feature detection and matching in SIFT as in [38] are represented with the following expressions for each pixel coordinates x and y:

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

where, $G(x, y, \sigma)$ is the Gaussian kernel with standard deviation σ, $*$ denotes convolution and $I(x, y)$ is the original image. A difference of Gaussians (DoG) is calculated as follows:

$$D(x, y, \sigma) = G(x, y, k\sigma) - G(x, y, \sigma), \tag{2}$$

where k is a constant factor that determines the ratio of scales between adjacent levels of the scale-space

$$D(x) = D + \frac{1}{2}\frac{\partial D^T}{\partial x}x + \frac{1}{2}x^T\frac{\partial^2 D}{\partial x^2}x, \tag{3}$$

where $D(x)$ is the Taylor expansion of the DoG function at the key-point location x. The first derivative and Hessian matrix of the DoG function are computed at x to obtain the expansion coefficients [39], magnitude $m(x, y)$ and orientation $\theta(x, y)$

$$m(x, y) = \sqrt{L_x^2 + L_y^2}, \theta(x, y) = \tan^{-1}\frac{L_y}{L_x}. \tag{4}$$

Here, L_x and L_y are respectively the horizontal and vertical components of the local image gradient at point (x, y). The magnitude and orientation of the gradient are computed to construct the gradient orientation histogram.

The i-th element of the SIFT descriptor vector, which consists of 128 elements and is denoted as d_i and can be expressed as

$$d_i = \sum_{x,y}\omega(x, y)h_i(x, y), \tag{5}$$

with $h_i(x, y)$ being the histogram of gradient orientations within a sub-region centred on the key point. The $\omega(x, y)$ weighting function gives high weight to gradients close to the key-point centre.

Speeded-up robust features (SURF) were presented in Bay's paper [40]. The author of that paper proposes a new descriptor that is superior to methods that are considered to be state-of-the-art. This descriptor is based on sums of Haar wavelet components, and the author believes that it is a scale and rotation-invariant interest point detector and descriptor that is both quick and effective. In addition to this, the description has superior performance to the histogram-based techniques that are utilised in the SIFT method.

The basic processes of SURF are roughly the same as SIFT [32]. In Bay's paper [33], there are three major processes to finding discrete image point correspondences. The first step is selecting the 'interest points like corner and T-junction, then calculating a feature vector to represent the neighbourhood of every selected interest point. Finally, the feature vectors of different images are matched.

Unlike SIFT, which is based on integral images for image convolutions, SURF, which is provided in Bay's article [33], employs a detector that is based on the Hessian filter rather than a detector that is based on a histogram of locally oriented gradients near the key point. Because box filters and integral pictures are being used, the filter size is being scaled up rather than scaled down while moving between different scale-spaces. This is done in place of the SIFT technique of down-sampling. Bay [33] is of the opinion that this adjustment might result in an increase in the computational output.

SURF is more efficient with respect to performance matching speed in comparison to SIFT [33]. In the area of matching images, therefore, Liu [41], Verma [42] and Vardhan [43] used SURF and verified that this approach has high performance and robustness. In regard to the performance of distorted image matching, Karami [44] compares the performance of SURF and SIFT, and the author denotes that SIFT has a higher matching rate in most rotation angles than SURF. Fields that SURF can be used based on the computation performance that offers are, the SAR image matching [41], image registration [45], visual tracking [46] and face recognition [47].

C. Object Tracking

Every frame is processed with SIFT algorithm for determining the matching key points between the prior knowledge and the current frame. This can lead to higher computation time. The usage of tracking algorithms on the region of interest that was detected is used to reduce the computation. The object of interest is often detected and the region of interest is defined as a bounding box. The object of interest can be detected manually, by a human operator or automatically, and it is often a UAV, a wild animal or anything else in a series of frames.

This paper proposes two methods for runway detection which include two parts - first runway shape detection with the SIFT algorithm by detecting key points and next using the Channel and Spatial Reliability Tracker (CSRT) algorithm [48] that uses pixel speed in the image plane and detects the bounding box in the consecutive video frames.

The tracking models are described as the motion model which tracks the speed and the direction of the object movement and the appearance model which is the object in the frame [49]. The library that is being used for deriving the tracking algorithms is OpenCV. The package consists of built-in functions for object tracking, and different trackers are available to be used as proposed by Rublee et al. [50]. The algorithm that is chosen to be used for the tracking application is the CSRT tracker as it is accurate for the purpose of runway tracking after detecting the region of interest. Alternative trackers are the Boosting Tracker [51] and Kernelized Correlation Filters [52].

A comparative study was presented in [53], analysing the different available tracking algorithms and denoting the efficiency of the CSRT tracker that was used for the development of the system.

The next Section IV describes the developed methodology followed for runway identification.

IV. Methodology

A. Proposed Architecture

The methodology that is at the core of the proposed real-time approach is based on the SIFT and SURF feature detectors and matching [33] that is presented in Section III. The developed system is a combination of feature matching between the current frame from the simulation environment and tracking of the detected box as described in section III.

For the data generation of the flight scenarios, the X-Plane 11 simulator was used. The simulation environment allows the user to control the plane from take-off to landing. However, for the purpose of the experiments for the automatic landing using a vision-based system, an autopilot was used, in order to set the mission of the plane that is taking off from the airport, executing a circle around and landing at the airport. This makes video processing efficient and reduces computational time.

The network constructed is using a host machine that is acting as the UAV (flight simulator), with the autopilot module and the mission. A recording script was developed to extract the frames from the simulator window and transmit the data to the processing unit for image processing. Flight information is made available to the processing unit by Distributed Avionics' Distributed Control software [54]. All flight information such as position, poise, mission plan and current waypoint is available this way. Those data enable the initiation of the algorithm on the landing approach. The post-processing data (runway detection) and the landing decision are visualised for the user.

When the landing procedure is initiated by the autopilot the Computer Vision system is notified to start the runway identification. To determine if a static point is inside a bounding box, a comparison between the point coordinates with the boundaries of the box is performed to check if they fall within the x and y ranges of the box. After the examination, a flag is increased, if the point is inside the bounding box, and the flag is decreased, if the point is outside the bounding box and heading off-track.

The static point is a result of the fixed camera location on the plane. The trajectory of the plane in order to land needs to be a straight line. This point is representing a point of the straight line in the image plane. Based on the flag value the decision of landing or abort is sent to the autopilot.

In order to reduce the computation time of the overall system, a tracking algorithm (CSRT) is used, that is tracking the speed and the direction of the detected runway in the next frames. After some amount of frames that the tracking algorithm used, a refinement of SIFT detection is executed.

B. *Software-In-The-Loop Simulation*

This section describes the operation of the software in the loop in order to enable the autopilot module to communicate with the X-Plane 11 simulator for navigating the plane in the simulation environment. The main processes of software in the loop system (SITL) are shown in Figure 4. After the mission is uploaded to the autopilot module the flight path is loaded and the plane is automatically executing the mission. A network is contracted using SITL for extracting and processing the images from the flight in-real time.

The advantage of using SITL for experimental purposes is testing and deployment. SITL is easy to deploy on a host machine and there is no need for other hardware equipment such as a real UAV or control system. Preliminary tests are run in different weather scenarios such as rain, wind, and fog.

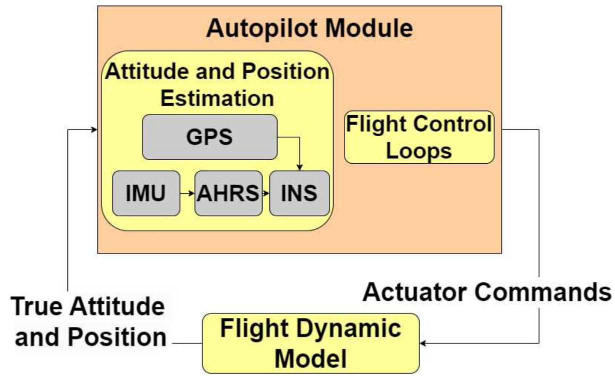


Fig. 4: Software-In-The-Loop (SITL) block diagram

C. *Feature Matching Results*

The proposed system is based on, feature matching using SIFT algorithm. SIFT is robust to affine distortion, changes in illumination, and moderate changes in viewpoint, making it a powerful tool for object detection in various scenarios. The Computer Vision algorithm is enabled when the autopilot sends the information about the current way-point being the landing way-point. When the landing way-point information is received the system enables the vision-based module. The module is searching for the runway using reference images from prior knowledge.

The prior knowledge is extracted by using prior flight footage. The user of the Computer Vision system is extracting the runway images from the prior knowledge and saves the reference images to a directory to be used by the vision-based system.

The prior knowledge is enabling the identification of the runway in the current frame. The key points and the descriptors of the reference images are extracted before the module starts. The reference image that is being used for feature matching is selected based on the number of matches found between the current frame and reference. The reference image used for bounding box extraction is selected based on the highest rate of matches between the reference and the current frame.

The resultant bounding box of the runway is used as input for the tracker algorithm. The tracking algorithms have less computation time than the feature detection and matching between the reference and the current frame. The algorithm is then used for a certain amount of frames to be executed on. A refinement of detection and matching is executed after the algorithm tracked the bounding box in a fixed number of frames, in order to provide accurate results.

The decision for landing or abort is taken by the system after the execution of feature matching and tracking. The decision-making is based on counters that are monitoring that the static point is enclosed in the bounding box detected. When the static point counter is negative, the decision is aborted as the UAV is heading off-track.

D. *Matching Points Extraction*

The system works by detecting key points in an image and then describing each key point, which is a set of features that describe the local region around the key point. SIFT uses a scale-space extrema detection to find distinctive features, which are invariant to image scale and rotation changes. Once the features have been detected, the matching between images is performed, which allows object detection and recognition.

The image-matching process is between a reference image that was derived from prior flights of the UAV in the same environment and frames from the current flight. The proposed system is extracting the overall number of matches that the algorithm finds. The matches threshold is used as a condition to specify good detection and matching results are observed.

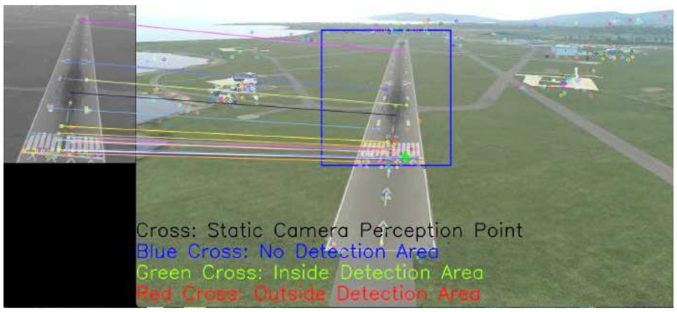


Fig. 5: Feature matching results on simulation environment

Furthermore, after a good observation, the bounding box is generated in order to pass the information to the tracking algorithm. The matching results between the reference image and the current frame can be observed in Figure 5.

In order to test the invariance of the feature matching algorithm on rotation, the reference image in Figure 5 is rotated. Figure 6, shows the bounding box placed in the original video frames after detection.

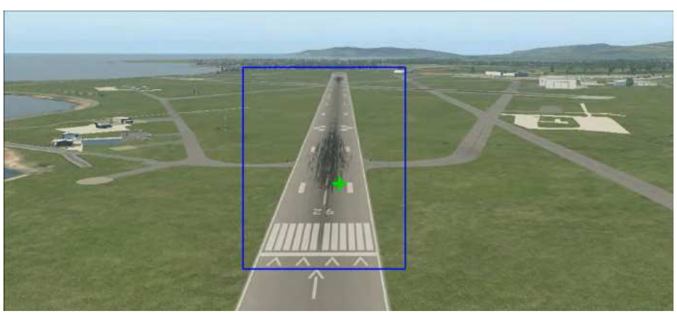


Fig. 6: Detection results and visualisation

V. Performance Evaluation

The proposed classification approach was used to evaluate the system for automated UAV landing. After the runway is identified in the frame a class is assigned to the frame as 1, and 0 when no runway is detected. The evaluation process can be observed in Figure 7. 2D(Dimensional) distances between the UAV and the landing point were calculated to measure the accuracy of the classification results, in distance ranges, for testing detection on runways observed in further/closer distances with respect to the UAV in the frames.

The Haversine distance is a mathematical equation 6 that is used to calculate the great-circle distance between two points on a sphere, such as the Earth. The formula uses the latitude and longitude of the two points and the radius of the sphere to calculate the central angle between the points [55]. The great-circle distance is then calculated as the product of the central angle and the sphere's radius, as follows

$$d = 2r\sin^{-1}\left(\sqrt{D_{Lat} + \cos(Lat_1)\cos(Lat_2)D_{Lon}}\right). \quad (6)$$

Here d is the distance between the two points, r is the radius of the sphere (such as the Earth's radius) and has a constant value of 6,378 km, Lat_1 and Lat_2 are the latitudes, and Lon_1 and Lon_2 are the longitudes of the two points. The distance between the values of latitudes and longitudes are as $D_{Lat} = \sin^2\left(\frac{Lat_2 - Lat_1}{2}\right)$ and $D_{Lon} = \sin^2\left(\frac{Lon_2 - Lon_1}{2}\right)$ respectively.

The evaluation methodology is based on literature and the FAA standards [56] for automatic UAV landing [17], [57]. Figure 8, refers to the evaluation of the system and the category distances are in meters. The reduction in accuracy for the nearest ground distance is due to the reference images depicting the closest view of a runway. Table I presents the accuracy results from different scenarios. Such scenarios are 1) clear weather conditions, 2) difficult weather conditions such as rain, or storm, and 3) cross landing conditions.

The ground truth data is generated from the results of detection and classification tasks which account for the 2D distance between the UAV and the landing point. The detection is performed before classification using prior knowledge. The classification task assigns a value equal to one to frames where the runway is detected and the landing is initiated. The 2D distances are assigned on frames after the autopilot enables the vision-based system. The system makes a decision at a certain altitude.

The runway detection algorithm was able to correctly identify the class in an average of 94.89% from videos generated and the average processing time for feature matching or tracking on a frame is 0.23s. Table I gives the accuracy of the runway identification in the specific distance ranges.

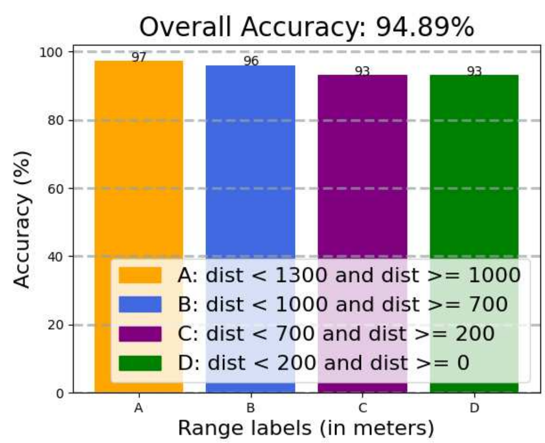


Fig. 8: Clear weather scenario evaluation results

VI. Conclusions

Motivated by the necessity of providing a high level of autonomy for UAVs, this paper develops a real-time vision-

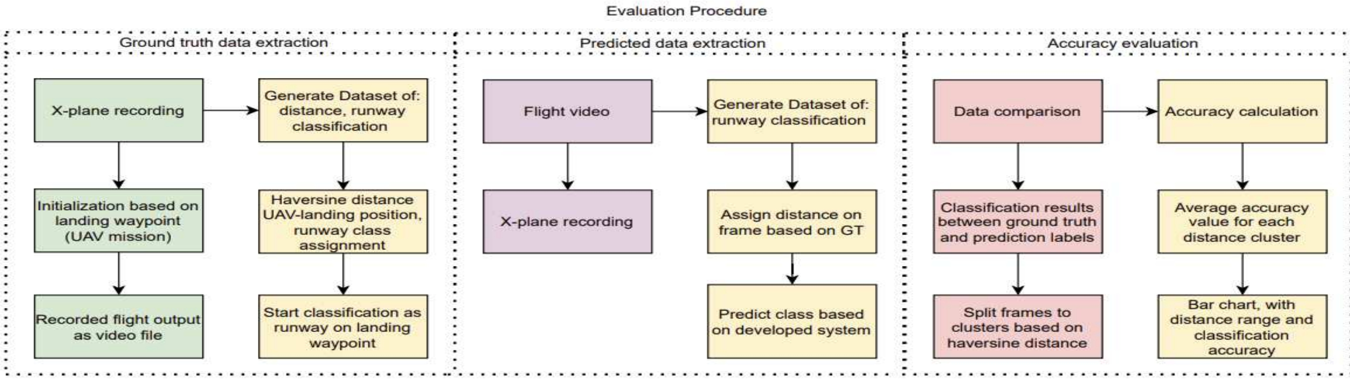


Fig. 7: Evaluation procedure block diagram

TABLE I: Accuracy of the proposed architecture

Distance Range	Scenario		
	Clear weather	Difficult weather	Cross landing
A:1300m - 1000m	100 %	99 %	98 %
B:1000m - 700m	100 %	95 %	93 %
C: 700m - 200m	91 %	90 %	90 %
D: 200m - 0m	93 %	93 %	89 %

based system for runway detection and UAV landing. We first reviewed the recent related Computer Vision methods and discussed their pros and cons. Deep learning methods are in their early stage of development, require a lot of data for training and this still creates obstacles to their applications on UAVs. Real-time solutions can be achieved with a combination of computationally efficient Computer Vision methods able to learn from the data. In this work, the developed multi-image matching system achieves both real-time and accurate runway detection by embedding SIFT, SURF and the CSRT tracking algorithms all implemented on a Jetson Xavier NX board. The results of simulated data show that the average accuracy is 94.89% in clear weather conditions. The performance validation shows a real-time image processing speed of 0.23 seconds on average for a frame. Future work will focus on introducing learning methods able to achieve real-time performance such as transfer learning and approaches for difficult weather conditions.

VII. Acknowledgment

We are grateful to the support from the Innovate UK, via grant from the Future Flights 3 strand: "Protecting environments via a swarm of UAVs". For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) license to any Author Accepted Manuscript version arising.

References

[1] Windracers, "ULTRA unmanned aerial vehicle." https://windracers.org/, 2023.

[2] R. Perz and K. Wronowski, "UAV application for precision agriculture," *Aircraft Engineering and Aerospace Technology*, vol. 91, pp. 257–263, 2018.

[3] L. Zongjian, "UAV For Mapping — Low Altitude Photogrammetric Survey," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVII, Part B1, Beijing, p. 4, 2008.

[4] D. Harel, A. Marron, and J. Sifakis, "Autonomics: In search of a foundation for next-generation autonomous systems," *Proceedings of the National Academy of Sciences*, vol. 117, pp. 17491–17498, July 2020. Company: National Academy of Sciences Distributor: National Academy of Sciences Institution: National Academy of Sciences Label: National Academy of Sciences Publisher: Proceedings of the National Academy of Sciences.

[5] H.-M. Huang, "Autonomy Levels for Unmanned Systems (ALFUS) Framework vol. I: Terminology version 2.0," 2004.

[6] N. J. McNeese, M. Demir, N. J. Cooke, and C. Myers, "Teaming With a Synthetic Teammate: Insights into Human-Autonomy Teaming," *Human Factors*, vol. 60, pp. 262–273, Mar. 2018. Publisher: SAGE Publications Inc.

[7] H.-M. Huang, "Autonomy Levels for Unmanned Systems (ALFUS) Framework: Safety and Application Issues," in *Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems*, (New York, NY, USA), p. 48–53, ACM, 2007.

[8] K. Abu-Jbara, W. Alheadary, G. Sundaramorthi, and C. Claudel, "A robust vision-based runway detection and tracking algorithm for automatic UAV landing," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1148–1157, 2015-06.

[9] Z. Daixian, "SIFT algorithm analysis and optimization," in *Proceeding of the 2010 International Conference on Image Analysis and Signal Processing*, pp. 415–419, 2010. ISSN: 2156-0129.

[10] E. Abbott and D. Powell, "Land-vehicle navigation using GPS," *Proceedings of the IEEE*, vol. 87, no. 1, pp. 145–162, 1999.

[11] A. Patrik, G. Utama, A. A. S. Gunawan, A. Chowanda, J. S. Suroso, R. Shofiyanti, and W. Budiharto, "GNSS-based navigation systems of autonomous drone for delivering items," *Journal of Big Data*, vol. 6, 2019.

[12] J. Akbar, M. Shahzad, M. I. Malik, A. Ul-Hasan, and F. Shafait, "Runway detection and localization in aerial images using deep learning," in *Proc. of the Digital Image Computing: Techniques and Applications (DICTA) Conference*, (Perth, WA, Australia), pp. 1–8, 2019.

[13] M. E. Cantoni Virginio, *Hough Transform*. New York: Springer, 2013.

[14] C. Liu, I. Cheng, and A. Basu, "Real-Time Runway Detection for Infrared Aerial Image Using Synthetic Vision and an ROI Based Level Set Method," *Remote Sensing*, vol. 10, no. 10, 2018.

[15] Z. Yang, J. Zhou, and F. Lang, "Detection algorithm of airport runway in remote sensing images," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 12, 2014.

[16] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[17] S. Nazir, S. Aziz, Y. Khaliq, and S. M. Adnan, "Vision based autonomous runway identification and position estimation for uav landing," in *Proc. of International Conference on Artificial Intelligence and Data Processing (IDAP)*, pp. 1–6, IEEE, 2018.

[18] F. Chen, R. Ren, T. Van de Voorde, W. Xu, G. Zhou, and Y. Zhou, "Fast automatic airport detection in remote sensing images using convolutional neural networks," *Remote Sensing*, vol. 10, no. 3, 2018.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. of the 28th International Conference on Neural Information Processing Systems*, vol. 10, pp. 91–99, 2015.

[20] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: a Line Segment Detector," *Image Processing On Line*, vol. 2, pp. 35–55, 2012.

[21] P. Zhang, X. Niu, Y. Dou, and F. Xia, "Airport detection on optical satellite images using deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 8, pp. 1183–1187, 2017.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, (Red Hook, NY, USA), p. 1097–1105, Curran Associates Inc., 2012.

[23] A. B. Sargano, P. Angelov, and Z. Habib, "A comprehensive review on handcrafted and learning-based action representation approaches for human activity recognition," *Applied Sciences*, vol. 7, no. 1, p. 110, 2017.

[24] K. Litjens, G. Ghafoorian, J. van der Laak, B. van Ginneken, and A. Roselaar, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[25] T.-Y. Lin, P. Doll'ar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, Cham, 2014.

[26] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[27] O. A. Ogunmolu and O. O. Oguntoyinbo, "Object detection with deep learning: A review," *arXiv preprint arXiv:1910.09702*, 2019.

[28] X. Yuan, H. Ma, and X. Hu, "Deep Learning for Object Detection: A Comprehensive Review," *IEEE Access*, vol. 8, pp. 168737–168745, 2020.

[29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Deep Learning in Computer Vision: A Review," *arXiv preprint arXiv:1804.09569*, 2018.

[30] T. Krajacic, "Implementing X-Plane as a Visual System for a Research Flight Simulator," Master's thesis, Institute of Mechanics, Graz University of Technology, 2012.

[31] A. Bittar, H. V. Figuereido, P. A. Guimaraes, and A. C. Mendes, "Guidance Software-In-the-Loop simulation using X-Plane and Simulink for UAVs," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014.

[32] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[33] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[34] M. Sharifi, M. Fathy, and M. Mahmoudi, "A classified and comparative study of edge detection algorithms," in *Proceedings of the International Conference on Information Technology: Coding and Computing*, pp. 117–120, 2002.

[35] X. Hu, Y. Tang, and Z. Zhang, "Video object matching based on SIFT algorithm," in *Proceeding of the 2008 International Conference on Neural Networks and Signal Processing*, pp. 412–415, 2008.

[36] W. Zhang, "Combination of SIFT and Canny Edge Detection for Registration Between SAR and Optical Images," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022. Conference Name: IEEE Geoscience and Remote Sensing Letters.

[37] A. Miller, M. Shah, and D. Harper, "Landing a UAV on a runway using image registration," in *Proceeding of the 2008 IEEE International Conference on Robotics and Automation*, pp. 182–187, 2008.

[38] D. Lowe, "Object Recognition from Local Scale-Invariant Features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157, 1999.

[39] U. Shah, D. Mistry, and A. Banerjee, "Image Registration of Multi-View Satellite Images Using Best Feature Points Detection and Matching Methods from SURF, SIFT and PCA-SIFT," *Journal of Emerging Technologies and Innovative Research*, vol. 1, no. 1, pp. 8–18, 2014.

[40] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *Computer Vision – ECCV 2006* (A. Leonardis, H. Bischof, and A. Pinz, eds.), Lecture Notes in Computer Science, pp. 404–417, Springer, 2006.

[41] R. Liu and Y. Wang, "SAR Image Matching Based] on Speeded Up Robust Feature," in *Proceeding in 2009 WRI Global Congress on Intelligent Systems*, vol. 4, pp. 518–522, 2009.

[42] N. K. Verma, A. Goyal, A. H. Vardhan, R. K. Sevakula, and A. Salour, "Object Matching Using Speeded Up Robust Features," in *Intelligent and Evolutionary Systems* (K. Lavangnananda, S. Phon-Amnuaisuk, W. Engchuan, and J. H. Chan, eds.), Proceedings in Adaptation, Learning and Optimization, pp. 415–427, Springer International Publishing, 2016.

[43] A. H. Vardhan, N. K. Verma, R. K. Sevakula, and A. Salour, "Unsupervised approach for object matching using Speeded Up Robust Features," in *Proceeding of the 2015 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pp. 1–8, 2015. ISSN: 2332-5615.

[44] E. Karami, S. Prasad, and M. Shehata, "Image matching using SIFT, SURF, BRIEF and ORB: Performance comparison for distorted images," in *Proceedings of the Newfoundland Electrical and Computer Engineering Conference*, 2015.

[45] R. Bouchiha and K. Besbes, "Automatic Remote-sensing Image Registration Using SURF," *International Journal of Computer Theory and Engineering*, 2013.

[46] J. Li, Y. Wang, and Y. Wang, "Visual tracking and learning using speeded up robust features," *Pattern Recognition Letters*, vol. 33, no. 16, pp. 2094–2101, 2012.

[47] S. Gupta, K. Thakur, and M. Kumar, "2D-human face recognition using SIFT and SURF descriptors of face's feature regions," *The Visual Computer*, vol. 37, no. 3, pp. 447–456, 2021.

[48] A. Lukežič, T. Vojíř, L. Č. Zajc, J. Matas, and M. Kristan, "Discriminative Correlation Filter Tracker with Channel and Spatial Reliability," *International Journal of Computer Vision*, vol. 126, no. 7, pp. 671–688, 2018.

[49] A. Sarkar, S. Negi, and A. Dangi, "Comparison of different tracking algorithms in OpenCV," *Ijraset Journal For Research in Applied Science and Engineering Technology*, vol. 10, pp. 596–598, 2022.

[50] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2564–2571, 2011.

[51] H. Grabner, M. Grabner, and H. Bischof, "Real-Time Tracking via Online Boosting," vol. 1, pp. 47–56, 2006.

[52] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-Speed Tracking with Kernelized Correlation Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[53] V. Agarwal, N. Sivakumaran, and D. V. Naidu, "Six Object Tracking Algorithms: A Comparative Study," *Indian Journal of Science and Technology*, vol. 9, 2016.

[54] D. Avionics, "Distributed avionics specialises in high-reliability flight control solutions for drone platforms." https://distributed-avionics.com/, 2023.

[55] T. Vincenty, "Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations," *Surveys in geophysics*, vol. 1, no. 2, pp. 75–116, 1975.

[56] F. A. Administration, "Neural network based runway landing guidance for general aviation autoland," Technical Report, Federal Aviation Administation, 2021.

[57] A. J. Moore, M. Schubert, C. Dolph, and G. Woodell, "Machine Vision Identification of Airport Runways with Visible and Infrared Videos," *Journal of Aerospace Information Systems*, vol. 13, pp. 266–277, 2016.