Vision-based UAV Safe Landing exploiting Lightweight Deep Neural Networks

CHARALAMPOS SYMEONIDIS, EFSTRATIOS KAKALETSIS, IOANNIS MADEMLIS, NIKOS NIKOLAIDIS, ANASTASIOS TEFAS, IOANNIS PITAS

Department of Informatics, Aristotle University of Thessaloniki, Greece

Email: {nnik, tefas, pitas}@csd.auth.gr

Recent advances in artificial intelligence, control and sensing technologies have facilitated the development of autonomous Unmanned Aerial Vehicles (UAVs, or drones) able to self-navigate in various settings. Although these technologies have already entered a mature stage, ensuring flight safety in crowded areas or performing an emergency landing in case of malfunctions, while adhering to relevant legislation, is generally treated as an afterthought when designing autonomous UAV platforms for unstructured environments. This paper proposes a UAV safe landing navigation pipeline that relies on lightweight computer vision modules, able to be executed on the limited computational resources on-board a typical UAV. Pre-trained Deep Neural Networks (DNNs) are mainly employed as the underlying building blocks, since deep learning has made a major impact on robotic perception by drastically improving the performance of relevant tasks, such as object detection or tracking, semantic image segmentation, etc. Evaluation of the proposed pipeline on a simulated environment indicates highly favorable results.

CCS CONCEPTS • Computing methodologies ~ Machine learning ~ Machine learning approaches ~ Neural networks • Computer systems organization ~ Embedded and cyber-physical systems ~ Robotics

Keywords and Phrases: UAV safe landing, robot navigation, object detection, semantic image segmentation, path planning, autonomous drones

1 INTRODUCTION

Technological progress has led to an increasing use of autonomous vehicles, with complex signal processing, computer vision and machine learning algorithms facilitating their operation. In most cases, such a vehicle must accurately perceive its environment at all time instances, so as to successfully navigate while avoiding obstacles, detect/track its targets, etc. Among the various types of autonomous/semi-autonomous vehicles, Unmanned Aerial Vehicles (UAVs), or drones, have attracted considerable attention, since they have proven useful for many civilian and military applications, such as surveillance, search and rescue operations, inspection, mapping, as well as in media production [1, 2].

Flight safety is of the utmost importance in autonomous UAVs, due to the hazard they pose in case of malfunction for people and equipment. In fact, any autonomous system must ensure safety in its interactions with the environment, especially when operating near humans. This is usually achieved through several interacting building blocks, each based on appropriate algorithms and sensors. Computer vision methods are

at the forefront of this ecosystem, due to the wide availability of visual sensors and the high progress in image/video analysis during the past decade. For instance, visually identifying potential landing areas, which in general should be sufficiently flat, large and unoccupied by people, vegetation or buildings/cars/water, is important for normal and emergency landings alike. Both should be carried out safely, avoiding people and without causing human injuries or damaging third parties. However, current autonomous UAV platforms do not provide such functionalities in a systematic manner.

One reason for this gap is that algorithms running on UAVs face more stringent limitations compared to other autonomous systems, arising from their bounded computational power due to weight and battery power constraints. As a result, developing relevant algorithms able to efficiently run on-board a UAV is challenging. For instance, Deep Neural Networks (DNNs) achieving state-of-the-art results in computer vision tasks typically require a lot of computational resources for real-time operation.

Thus, existing UAV safe landing pipelines are not able to visually confirm safe landing sites in real-time using camera footage and rely only on a-priori available terrain geometry information, which is not always up-to-date or complete (e.g., due to varying environment conditions, dynamic obstacles, presence of impermanent water bodies, etc.). Where video footage is indeed analyzed [3,4,5,6,7], only landing suitability and danger of navigability are determined, instead of fully extracting (much richer) visible scene semantics information.

In this paper, a vision-based autonomous UAV safe landing pipeline is proposed that is built upon deep, but lightweight neural modules, able to be reliably executed on-board a UAV in an on-line fashion. The described approach mainly exploits algorithms for three vision-based tasks: (a) landing-site detection, (b) person/human detection, (c) semantic image segmentation. An additional path planning module exploits their outcomes and ensures safe navigation for the autonomous vehicle, as empirically indicated in Section 4. Unlike existing approaches, the proposed pipeline extracts on-the-fly and stably rich scene semantic information from video footage, which is exploited to maximize autonomous UAV flight and landing safety.

2 RELATED WORK

2.1 Landing site Detection

Areas suitable for landing can be detected by utilizing visual information (e.g., videos from the UAV camera) or map/3D terrain information in the form of well-known formats such as Octomap [8] or Digital Elevation Models (DEM). Literature referring to landing site detection is rather limited. In [9] a UAV landing site detection system using mid-level sub-image discriminant patches is described. Image patch detectors are trained and tuned with several images obtained from Google Maps, weakly labeled as “Very dangerous", “Not Recommended", or “Safe". This way, a heatmap of the examined video frame is created. The final landing sites are defined by thresholding the heatmap. Moreover, an aerial image classification system for a UAV forced landing site detection task is proposed in [10] for classifying image segments via a Gaussian Mixture Model (GMM) or an SVM into two categories, “safe" or “unsafe". Utilizing terrain maps, [11] detects landing sites for fixed-wing UAVs in emergency situations by using quadtree-based DEM partitions. Furthermore, an image processing-based method for UAV potential landing site detection using a-priori terrain information through identification of flat areas on available DEMs, i.e., on Digital Terrain Models (DTMs) and Digital Surface Models (DSMs), is presented in [12].

2.2 Deep learning-based person/human detection

An important part of autonomous UAV flight safety is person/human detection based on video input captured on-the-fly. UAV safe landing requires that the UAV visually detects any individuals [13] around the landing area, in order to avoid harming them in case of a malfunction; airspace above/near humans should be considered as a no-fly zone. Such detectors assign a discrete class label (out of *K* pre-specified object classes) to a an appropriately sized rectangular bounding box (Region-of-Interest, or ROI) surrounding each object detected on the image.

Early deep neural methods [14,15] on person detection employed CNNs based on the R-CNN [16] object detection architecture, which relies on high-quality externally provided proposals to achieve good performance. Recent works [17] are using single-stage detectors, e.g., YOLOv2 [18]. The end-to-end nature of these detectors results in significantly increased speed, although accuracy slightly suffers in comparison to Faster R-CNN [34]. In [19] the authors address the impact of the small size of pedestrians and the deformation of camera's perspective projection in YOLO's overall accuracy and try to overcome these problems by using a combination of cropping-resizing and projective geometry-based vanishing point transformation. RetinaNet [20] is another single-stage detector that shows comparable detection performance with two-stage approaches. It uses a Feature Pyramid Network [21] as backbone, on top of a ResNet architecture. Two disjoint sub-networks respectively classify anchor boxes and adjust the values with respect to the default anchors. Finally, in [22] a Non-Maximum Suppression neural network is modified and coupled with various neural detectors for handling aerial person detection tasks.

2.3 Semantic image segmentation

Semantic segmentation consists in assigning a discrete class label (out of *K* pre-specified object classes) to each pixel of an input image. The field advanced significantly during the past decade, mainly by employing supervised deep neural models for semantic per-pixel image analysis. Semantic segmentation networks are typically composed of an encoding and a decoding subnetwork, arranged in a consecutive fashion. The encoder extracts from the input semantic features with progressively lower spatial resolution, while the decoder receives the final encoder output and upsamples it. The final output of the decoder is a dense semantic map, having the same spatial resolution as the input. Fully convolutional networks [23] and dilated convolutions [24] are typically used, for upsampling the computed abstract feature maps and for enlarging neuronal receptive fields, respectively. Large receptive fields significantly aid semantic segmentation by enriching local-scale image representation with task-relevant wider region semantic context, for more accurate per-pixel classification. Grasping global image context while retaining spatial detail is an important consideration in relevant research, with current DNNs attempting to explicitly capture it and properly enhance local image representation, using multi-scale [25], attention-based relational [26], or network branching approaches [27]. PSPNet [25] offers a good balance between speed and accuracy, forming also the backbone of more recent real-time segmentation networks. Its main novelty is a PPM (Pyramid Pooling Module) decoder, able to enrich local image representation with more global context information from larger image regions of various scales.

3 PROPOSED UAV SAFE LANDING PIPELINE

The proposed algorithmic pipeline assumes that the UAV is equipped with: (a) a flight controller, e.g., PixHawk/PX4 Autopilot, (b) a computing board (containing both a CPU and a GP-GPU), e.g., an NVIDIA Jetson Xavier, (c) a monocular RGB camera suspended from a gimbal that allows rapid, arbitrary rotation around its yaw, pitch and roll axis, as well as (d) localization sensors, e.g., a GPS sensor.

The proposed pipeline aims at ensuring a safe autonomous UAV landing process in case of emergency, relying on active environmental perception. It is designed as a collection of independent interacting modules, with easy inter-process communication and synchronization among them transparently assured by the popular underlying middleware Robotic Operating System (ROS) [28].

- The included modules are the following ones:
- Potential Landing Site Detection (PLSD)
- Visual Landing Site Detection Using Semantic Segmentation (VLSD)
- Person Detection (PD)
- Semantic Map Manager (SMM)
- Semantic Map Region Projector (SRP)
- Simple Path Planner (SPP)

3.1 Potential Landing Site Detection (PLSD)

The aim of this module is to identify (sufficiently) flat and large areas in topographical maps for safe UAV landing, in normal or emergency situations. The employed method [12] utilizes the information in pre-obtained Digital Terrain Model (DTM) and Digital Surface Model (DSM) files, in order to detect the vegetation, buildings and generally the objects upon the bare ground, by evaluating the height difference between the DTM and DSM models. Flat areas are discovered by evaluating the local terrain slope through estimating the image gradient on the DEM file and thresholding the gradient magnitude image, so as to retain areas having small local 3D gradient. Connected component analysis is applied on the resulting binary image, so as to identify and retain regions whose area is above a preset potential landing area threshold. The final map is constructed by combining the results of building and vegetation detection with the results of the previous step, to produce regions where landing shall not be attempted.

3.2 Visual Landing Site Detection Using Semantic Segmentation (VLSD)

Using semantic image segmentation, 2D image regions corresponding to ground areas that are most likely suitable for landing (such as grass, or pavement) can be identified and discriminated from areas unsuitable for landing (such as trees, people, water, or buildings). To this end, the popular PSPNet was adopted and properly adapted: instead of typically used, complex encoding/base feature extracting networks, we coupled the PSPNet network layers with a MobileNetV2 base feature extractor [23], enhanced with dilated convolutions [24], resulting in a relatively fast implementation at a low accuracy penalty.

The ADE20K semantic scene parsing dataset was selected for training [30]. It is a large dataset containing $K = 150$ distinct object classes from both indoors and outdoors environments, including all important classes relevant to the landing site detection task. To achieve a good accuracy/speed trade-off, during inference each input image is resized so that the short edge length (in pixels) is 180 pixels, while the aspect ratio is preserved

under the constraint that the longer edge length cannot be greater than 1000 pixels. Training was adjusted accordingly.

Additionally, the base feature extraction network was pruned by removing 5 intermediate convolutional layers, so as to speed-up execution during inference, resulting in a lightweight variant of the full MobileNetV2+PSPNet network. Finally, since the 150 discrete object classes of ADE20K are not necessary for potential landing site detection, they were manually clustered into 11 classes, relevant to outdoor UAV flight: Landing Area, Building, Background, Man-Made Structures, Vegetation, Sky, People/Animals, Vehicles, Light, Man-Made Objects, Water. Reducing the number of discrete classes simplifies the problem and allows the segmentor to generalize better.

3.3 Person Detection (PD)

The role of this module is critical for ensuring human safety. The detector should both perform accurately and run as close to real-time as possible. Thus, a dedicated detector was included in the pipeline, instead of employing the VLSD for solving this task as well through semantic image segmentation. In terms of the detector's accuracy, the small size of objects/persons (especially in high flight altitudes), as well as unforeseen and wide-ranging variations in illumination and camera orientation, are some of the main challenges. YOLOv2 [18] was selected as the proposed person detector; a fairly fast and computation-efficient solution.

3.4 Semantic Map Manager (SMM)

The Semantic Map Manager (SMM) is a module responsible for storing and updating the 3D map, which is annotated with landing sites and no-fly zones (human/person locations). In addition, the SMM is responsible for continuously checking whether the UAV's path towards a safe landing site remains eligible; a newly detected person may be too close to the path. In case the path is not eligible, the SMM requests a new path from SPP given the updated map annotations.

3.5 Semantic Map Region Projector (SRP)

This module is responsible for delineating 3D map (Octomap) regions obtained from the 2D-to-3D back-projection of image plane annotations, derived by PD/VLSD modules, using UAV position, gimbal orientation and camera parameters. As the UAV moves and its camera views new 3D terrain regions, the newly generated regions are merged with previously acquired ones, using the set union operator.

3.6 Simple Path Planner (SPP)

In order to compute an obstacle-free path to the preferred closest landing site, SPP receives as input the UAV's position along with geometric map annotations regarding the flight area limits, no-fly zones and landing sites.

Based on those, a 2D grid is formulated by ignoring the UAV's altitude; we assume that the UAV flies on a constant high altitude. The grid search is carried out by an A* algorithm [31], i.e., a heuristic search algorithm formulated for weighted graphs.

3.7 Putting it all together

The proposed vision-based safe landing pipeline is composed of the following stages:

3.7.1 Stage 1

The 3D map of the flight area (in Octomap form), along with the geo-referenced potential landing sites derived from PLSD, becomes available to the SMM. Computing both of them (off-line, before flight) requires the DSM and DTM height maps, which are publicly available with sufficient resolution for large parts of the globe. In a real-world scenario these maps may be outdated (e.g., an area's vegetation growth is highly volatile, even in a short time span).

3.7.2 Stage 2

During flight, the UAV constantly collects information about the presence of visible people (using the PD module), so as to avoid flying over them. However, once a safe landing spot is requested, the UAV must maintain a higher awareness of its environment as far as people are concerned. Thus, it periodically hovers and rotates the camera's gimbal at multiple pitch angles scanning for visible people located either far-away or near-by. This information initially takes the form of person ROIs (in pixel coordinates), which are on-the-fly transformed into 3D annotated regions/locations in geo-referenced 3D coordinates by the SRP. Then, they are stored on-line by the SMM along with the 3D area map, thus updating it with current semantic annotations.

3.7.3 Stage 3

When the need for landing arises, the SPP constructs on-line a route towards the closest approachable known safe landing zone, using the 3D semantic annotations projected onto a 2D map. The flight controller is ordered to follow the planned route and, as soon as the landing zone becomes visible via the camera, its actual eligibility for landing is evaluated using the VLSD module. At all times during this process a new route may be requested by the SPP, in order to: (a) avoid newly identified no-fly zones, consisting of air-space near/above people, or (b) select a different landing site (among the ones pre-computed by PLSD), if the originally selected one is not judged by the VLSD to be actually eligible.

3.7.4 Stage 4

As the UAV flies towards a safe landing zone, it may dynamically perceive a nearby people. If the vehicle's position at that moment lies within the no-fly zone defined by the people, the UAV cancels the current trajectory, dismisses the remaining part of the planned path and immediately flies to the closest point outside the no-fly zone. Once this point has been reached, a new path to that target landing site is requested by the SPP.

4 IMPLEMENTATION AND EVALUATION

To evaluate the proposed example UAV safe landing pipeline, a set of realistic simulations were created. A combination of AirSim [32] and ROS was used to that end. A large-scale (568x379m^2) realistic countryside environment (shown in Fig. 1) was populated with 80 people and employed for the evaluation process. The NVIDIA Jetson Xavier embedded AI compute board was selected for executing the overall pipeline.

Figure 1: Simulated AirSim environment.

A series of 15 independent missions (flight and landing) were simulated, using different randomly generated UAV starting points. In all cases, the UAV had access to the DTM and the DSM of the simulated environment. With the modules described above operating in synergy, the UAV would fly from its starting point towards the closest safe landing spot, avoiding no-fly zones and finally, land on it. In certain cases, the vehicle would have to on-line re-plan its path and land on a different landing site, since the original would be deemed as ineligible. The overall system was able to run steadily on NVIDIA Jetson Xavier with the most demanding modules, namely the PD and VLSD modules, achieving 9.5 FPS and 0.82 FPS respectively. The average flight time was 10.54 minutes approximately. In addition, the average number of times that a path was requested from SPP in a single mission was 1.53.

Performance of the proposed UAV safe landing pipeline was evaluated using the minimum lateral (on-ground projection) distance of the UAV trajectory waypoints from any person location in the terrain of the flight environment as a metric, as shown in Figure 3. High minimum distance ensures the avoidance of flying over humans and, thus, the overall safety of the autonomous landing process. Based on [33], the minimum legitimate lateral flight distance from individual people for several countries, including UK, Italy and Australia, is 30-50m. Thus, we set the minimum acceptable lateral distance to 50m. In our evaluation setup, relying on the vision-based detection of individuals from the PD in a scenario taking place at a mountainous environment populated with people, the mean minimum UAV-to-person distance along the actual UAV trajectory (averaged over all 15 independent missions) is 69.4637m, as shown in Figure 3 for several safe landing cases. Therefore, the minimum is exceeded on average by almost 20 meters.

The graph in Figure 3 depicts the (varying over time) minimum lateral distance of the UAV trajectory waypoints from people. The straight horizontal lines of the graph represent the time required for executing Stage 2 of the pipeline. Note that, rarely, the minimum drone-to-person distance falls below the minimum acceptable threshold for brief intervals (e.g., in mission 3). This may occur as people could not be detected until that point in time, e.g., due to visual occlusions. However, as soon as they are detected, the UAV automatically moves away from it (Stage 4), if necessary, and a new path is requested from SPP.

An example mission is visualized in Fig. 2. The UAV constantly detects people while it is flying and projects their 2D ROIs on the 3D navigation map as annotations. When it needs to land, it plans a trajectory to the closest a-priori known landing site utilizing the SPP and avoiding people. The trajectory consists in a smooth path of waypoints. A typical example of the pipeline's behavior in a challenging scenario is depicted in Subfigures 2a-2f. The UAV starts from Point A and the SPP constructs a path towards Point B (Fig. 2a). While

flying to Point B, the UAV detects a person in Point C. Thus, it cancels the current path, dismisses the remaining part of the planned trajectory and immediately moves out of the newly annotated no-fly zone (dark-blue area around Point C). This motion is depicted as a red line from Point D to Point E (Fig. 2b). Upon reaching Point E, the SPP is asked to construct a new path towards Point F (Fig. 2c). Shortly before reaching

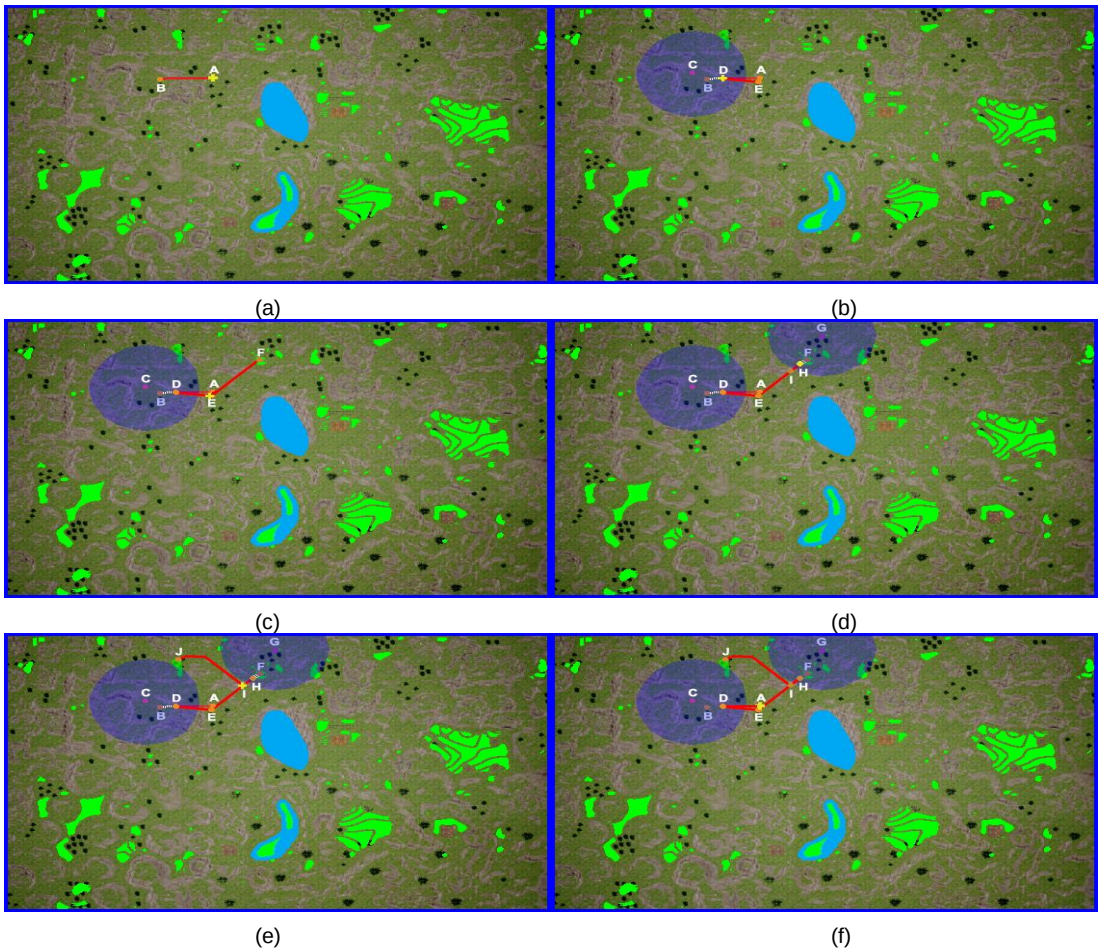

Figure 2: (a)-(e) UAV trajectory segments planned by the SPP at consecutive time intervals during a single mission; in each Subfigure, the UAV's location is marked as yellow "+", the pre-computed landing sites from Stage 1 are colored in light-green areas surrounded by water are highlighted as light-blue, the detected people are marked as purple dots, the detected no-flyzones are colored in dark-blue and the dismissed paths are depicted as a white-striped lines. (f) Actual UAV path [A, D, E, H, I, J] on the map during the entire mission.

Point F, the UAV detects a person in Point G and again flies away of the new no-fly zone it defines, cancelling once more the planned trajectory. This motion is depicted as a red line from Point H to Point I (Fig. 2d). Finally, Point J is determined as the closest safe landing spot and the SPP constructs a path towards it. This is depicted as a red line between Point I to Point J (Fig. 2e). Upon reaching Point J, VLSD confirms that it is indeed a safe landing spot and the UAV lands.

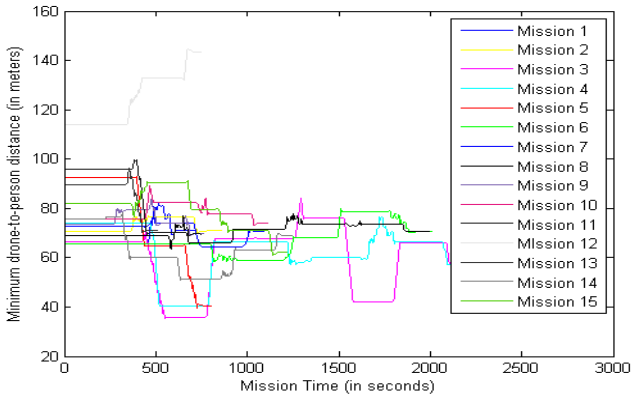

Figure 3: Minimum UAV-to-person distance (in meters) vs Mission time (in seconds).

5 CONCLUSION

UAVs with autonomous functionalities are increasingly being employed in various applications. Effectively ensuring human safety when deploying such autonomous systems involves a number of challenging tasks, which can be addressed successfully by using computer vision/machine learning methods based on DNNs running on-board the UAV and given camera input. To this end, a relevant autonomous UAV safe landing algorithmic pipeline is proposed, which mostly relies on vision methods for landing-site detection, person/human detection and semantic image segmentation. Through those methods, rich scene semantics are extracted aiming to maximize autonomous flight and landing safety. The pipeline was evaluated in simulated environments, using actual embedded AI computer hardware. Results indicate successful, real-time operation with automatic conformance to safety regulations. Future work will focus on incorporating and evaluating the proposed pipeline in a real-world scenario.

6 ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No. 731667 (MULTIDRONE) and No. 871479 (AERIAL-CORE). The authors would like to thank Yorgos Basioukas for his contribution in setting-up AirSim environments.

REFERENCES

[1] Mademlis, I., Mygdalis, V., Nikolaidis, N., & Pitas, I. (2018). Challenges in autonomous UAV cinematography: an overview. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME).

[2] Mademlis, I., Nikolaidis, N., Tefas, A., Pitas, I., Wagner, T., & Messina, A. (2018). Autonomous unmanned aerial vehicles filming in dynamic unstructured outdoor environments. IEEE Signal Processing Magazine, vol. 36, pp. 147–153.

[3] Mittal, M., Mohan, R., Burgard, W & Valada, A. (2019). Vision-Based Autonomous UAV Navigation and Landing for Urban Search and Rescue, arXiv preprint arXiv:1906.01304.

[4] Hinzmann, T., Stastny, T., Lerma, C.C., Siegwart, R. & Gilitschenski, I. (2018). Free LSD: Prior-free visual landing site detection for autonomous planes, IEEE Robotics and Automation Letters, vol. 3, pp. 2545-4552.

[5] Lee, M.-F. R., Aayush, J., Saurav, K. & Anshuman, D.A. (2020). Landing Site Inspection and Autonomous Pose Correction for Unmanned Aerial Vehicles, In Proceedings of the International Conference on Advanced Robotics and Intelligent Systems (ARIS).

[6] Demirhan, M. & Premachandra, C. (2020). Development of an Automated Camera-Based Drone Landing System, IEEE Access.

[7] Yang, T., Li, P., Zhang, H., Li, J. & Li, Z. (2018). Monocular Vision SLAM-Based UAV Autonomous Landing in Emergencies and Unknown Environments, Electronics, vol. 7, pp. 73.

[8] Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). Octomap: An efficient probabilistic 3D mapping framework based on octrees. Autonomous Robots, vol. 34, no. 3, pp. 189–206.

[9] Guo, X., Denman, S., Fookes, C., & Sridharan, S. (2016). A robust UAV landing site detection system using mid-level discriminative patches. In Proceedings of the International Conference on Pattern Recognition (ICPR).
[10] Guo, X., Denman, S., Fookes, C., Mejias, L., & Srid-haran, S. (2014). Automatic UAV forced landing site detection using machine learning. In Proceedings of International Conference on Digital Image Computing: Techniques and Applications (DICTA).
[11] Garg, M., Kumar, A., & P.B., S. (2015). Terrain-based landing site selection and path planning for fixed-wing UAVs. In Proceedings of the IEEE International Conference on Unmanned Aircraft Systems (ICUAS).
[12] Kakaletsis, E., & Nikolaidis, N. (2019). Potential UAV landing sites detection through digital elevation models analysis. In European Signal Processing Conference (EUSIPCO), Satellite Workshops.
[13] Kakaletsis, E., Tzelepi, M., Kaplanoglou, P. I., Symeonidis, C., Nikolaidis, N., Tefas, A. & Pitas, I. (2019). Semantic map annotation through UAV video analysis using deep learning models in ROS. In proceedings of the International Conference on Multimedia Modeling.
[14] Hosang, J., Omran, M., Benenson, R., & Schiele, B. (2015). Taking a deeper look at pedestrians. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[15] Zhang, S., Benenson, R., Omran, M., Hosang, J., & Schiele, B. (2016). How far are we from solving pedestrian detection? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[16] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[17] Lan, W., Dang, J., Wang, Y., & Wang, S. (2018). Pedestrian detection based on YOLO network model. In Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA).
[18] Redmon, J., & Farhadi, A. (2016). YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[19] Chang, Y.-C., Chen, H.-T., J.-H., C., & Liao, I.-C. (2018). Pedestrian detection in aerial images using vanishing point transformation and deep learning. In Proceedings of the IEEE International Conference on Image Processing (ICIP).
[20] Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2017). Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 2, pp. 318-327.
[21] Lin, T. Y., Dollar, P., Girshick, R., He, K., Hariharan B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[22] Symeonidis, C., Mademlis, I., Nikolaidis, N., & Pitas, I. (2019). Improving neural non-maximum suppression for object detection by exploiting interest-point detectors. In Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP).
[23] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[24] Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122.
[25] Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[26] Yuan, Y., Chen, X., & Wang, J. (2019). Object-contextual representations for semantic segmentation. arXiv preprint arXiv:1909.11065.
[27] Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., & Sang, N. (2018). Bisenet: Bilateral segmentation network for real-time semantic segmentation. In Proceedings of the European Conference on Computer Vision (ECCV).
[28] Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., . . . Ng, A. (2009). ROS: an open-source Robot Operating System. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) Workshop on Open Source Robotics.
[29] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[30] Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., & Torralba, A. (2017). Scene parsing through ADE20K dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[31] Duchoň, F., Babinec, A., Kajan, M., Beňo, P., Florek, M., Fico, T., & Jurišica, L. (2014). Path planning with modified a-star algorithm for a mobile robot. Procedia Engineering, vol. 96, pp. 59–69.
[32] Shah, S., Dey, D., Lovett, C., Kapoor, A., & Burgard, W. (2017). Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In Proceedings of the Field and Service Robotics Conference.
[33] Stöcker, C., Bennett, R., Nex, F., Gerke, M., & Zevenbergen, J. (2017). Review of the current state of UAV regulations. Remote Sensing, vol. 9, pp. 459.
[34] Ren, S., He, K., Gurshick, R. & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, Advances in Neural Information Processing Systems (NIPS), vol 28, pp 91-99.

Authors' Background

Name	Email	Position	Research Field
Charalampos Symeonidis	charsyme@csd.auth.gr	Ph.D. Student	Computer Vision, Machine Learning, etc.
Efstratios Kakaletsis	ekakalets@csd.auth.gr	Ph.D. Student	Computer Vision, Machine Learning, etc.
Ioannis Mademlis	imademlis@csd.auth.gr	Ph.D.	Computer Vision, Machine Learning, etc.
Nikos Nikolaidis	nnik@csd.auth.gr	Assoc. Prof.	Computer Graphics, Image and Video Processing, etc.
Anastasios Tefas	tefas@csd.auth.gr	Assoc. Prof.	Computational Intelligence, Deep Learning, etc.
Ioannis Pitas	pitas@csd.auth.gr	Prof.	Computer Vision, Machine Learning, etc.