Aalborg Universitet

Analyzing visual imagery for emergency drone landing on unknown environments

Bektash, O. M.; Pedersen, Jacob Naundrup; la Cour-Harbo, Anders

Analyzing visual imagery for emergency drone landing on unknown environments

Oghuz Bektash[1] [iD], Jacob Juul Naundrup[1] and
Anders la Cour-Harbo[2]

Abstract

Autonomous landing is a fundamental aspect of drone operations which is being focused upon by the industry, with ever-increasing demands on safety. As the drones are likely to become indispensable vehicles in near future, they are expected to succeed in automatically recognizing a landing spot from the nearby points, maneuvering toward it, and ultimately, performing a safe landing. Accordingly, this paper investigates the idea of vision-based location detection on the ground for an automated emergency response system which can continuously monitor the environment and spot safe places when needed. A convolutional neural network which learns from image-based feature representation at multiple scales is introduced. The model takes the ground images, assign significance to various aspects in them and recognize the landing spots. The results provided support for the model, with accurate classification of ground image according to their visual content. They also demonstrate the feasibility of computationally inexpensive implementation of the model on a small computer that can be easily embedded on a drone.

Introduction

Regulation for drones are becoming increasingly characterized by technology-facilitated applications as the drone use is rapidly expanding to serve in various applications of agriculture, scientific research, surveillance, search and rescue, infrastructure inspection and management, environmental monitoring and law enforcement.[1] Computational advancements and prodigious amount of condition monitoring data can be the key drivers of the future drone regulations, while the need for autonomous landing processes and growing acceptance of contemporary complex computer systems are also noted. Administrative authorities have been imposing restricting drone regulations with an attempt to secure safe and sustainable operations to assure public safety and privacy on condition that less restricted drone operations are available.[2,3] As these unmanned aircraft system regulations urge sustainable and secure drone operations to protect the people's safety, one can reasonably assume that the modern drones will have to be individually safe, allowing both legislation and end-users to provide drone reliability.

Previous research in the unmanned aircraft system (UAS) literature has introduced several frameworks such as computer vision aided positioning system,[4] autonomous tracking and landing control on an autonomous vehicle,[5] marker recognition on a landing pad,[6] vision analysis for automatic landing,[7] autonomous flight system with marker recognition,[8] or even illegal landfill detection.[9] Akbari et al.[10] reviewed applications related to drones and image recognition, and categorized them into various groups such as remote sensing, autonomous navigation and the sensed environment applications. As pointed out by them, the field still need to develop image recognition information for uncharted applications.

A major point of improvement between the drone safety and computational product development is the involvement

[1]Faculty of Engineering and Natural Sciences, Istanbul Medeniyet University, Turkey
[2]Automation and Control, Department of Electronic Systems, Aalborg University, Denmark

Corresponding author:
Oghuz Bektash, Faculty of Engineering and Natural Sciences, Istanbul Medeniyet University, Istanbul, Turkey.
Emails: oguz.bektas@medeniyet.edu.tr; oguz.bektas@warwickgrad.net

of analyzing visual landing location with fully connected multi-layer perceptrons. Particularly, convolutional neural networks (CNN) have received much attention in analyzing large-scale imagery which has recently become possible due to the advancing technologies.[11–15] In particular, there is a significant potential in the advance of CNN based visual recognition for drones, which can serve detecting locations of interest that can facilitate a landing, or even a crash, without any human injury or fatality. With CNN models becoming more of a commodity in the vision recognition,[15] they could let better compliance standards of drone activities by tracking safe landing locations and providing an easier development for operational process. It is therefore increasingly evident that these models will continue to be a critical component of drone situations, especially where the critical in-flight anomalies are experienced.

Drones have some characteristic safety drawbacks that are relatively existent in most UAS systems. Therefore, they need to be designed with a level of situational awareness and the "sense and avoid" capabilities that match those of manned aerial systems.[16] However, drones are usually under real-time human control with no on-board pilot and operated from a remote terminal. Therefore, their size result in a lower weight and a smaller extent than a peer manned aircraft, allowing only smaller embedded systems to be run. This means that, in spite of having a smaller on-board system computer, they must be able to respond with appropriate avoidance maneuvers to maintain safety[16] and to deal with risk perceptions that are generally elevated by ever increasing societal acceptance in company with the complex human behavior with machine interface and operational roles.[17]

For image recognition and visual recognition for autonomous drone landing, the unmanned systems in smaller sizes can only tolerate a small computer mounted on the vehicle which is also expected to track the health state and guidance for landing.[18,19] When there is a safety-critical condition, it is expected from this computer to determine the best course of action for the aircraft to minimize the probability of fatalities. This means a significant challenge for automated location recognition framework. Considering this issue in particular with CNN based visual recognition, the literature has not yet been clarified on the development of resource intensive application and its interface with the accompanying safety duties. Consequently, there is an incomplete picture of the way novel imagery landing location frameworks are developed. Against this background, the main purpose of this study is to investigate the role of CNN based visual landing recognition while considering the distinctive drone properties (see Figure 1). The study also aims to determine safety factors that may be associated with with the automatic location spotting.

To further understanding, this research has the following objectives:

- To find good landing spots that can facilitate an emergency landing or even a crash without casualties.
- To analyze a class of input image taken during an ongoing flight and output a class, or simply a probability of classes, that best describes whether the location is suitable for emergency landing.
- To develop a conceptual framework of a machine vision approach for detecting areas by outlining theoretical underpinnings and analyzing related research and construction schemes.
- To test the proposed model empirically by grabbing images from the camera mounted on drone (SafeEYE lab Bektash et al.[18]), which are monitored from various test flights where the primary focus was to provide images of a variety of terrains.

Because of the limited existence of on-flight pictures on previous research settings,[18] this study chose to further advance the autonomous location spotting framework with novel imagery with detailed surface scenes and examine whether CNN structure containing more visual information provide a strategic advantage for safety related operations. This means that this work has a further potential to become a safety enabler for a wide variety of drone applications.

The rest of the article is structured as follows: First, the extant literature on autonomous drone landing, vision recognition and convolutional neural networks are reviewed with an attempt to represent the theoretical core of the paper. This is followed by an introduction of the research methodology with procedures used in the work. A case study is then offered to analyze as the research is expected to read-through to particular context introduced in the methodology section earlier. The results of the research are then summarized with the findings of the study in the form of descriptive statistics. Finally, in the conclusion section, implications, limitations, and directions for future studies are laid out.

Background and Related Work

Recognition and detection of defined objects on captured scene images have been well-known challenges, even supposing that the literature have brought many solutions to existing problems. The growth of image recognition algorithms is a determining factor to overcome these issues. A number of works have been introduced to make picture processing studies possible on drone systems. Current strategies have used various designs such as using markers,[8] features classifiers and detectors,[20,7] and both static[21] and dynamic picture recognition.[22] Further works of autonomous drone landing can be regarded as a multifaceted construct consisting of various dimensions. Those of using image processing algorithms such as above-mentioned ones typically aims to find a spot to safely and securely
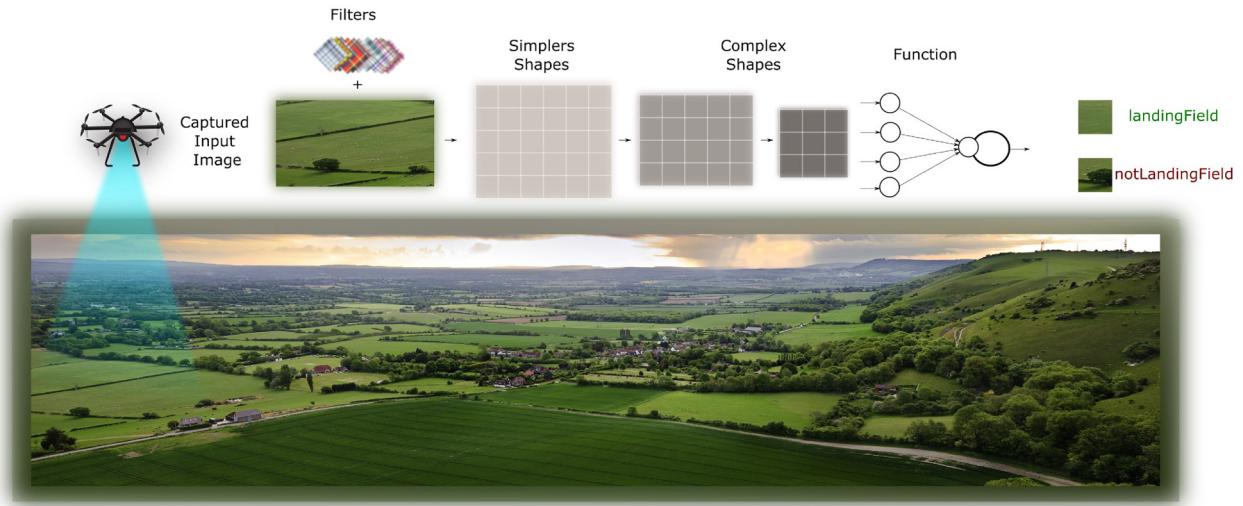
Figure 1. Graphical Abstract of Drone Image Processing.

land the drone. However, they should be accompanied with other autonomous actions such as malfunction detection and guidance for landing.[18] The explicit analyses of visual interpretation and image classification along with their relationship to these autonomous actions help provide evidence showing that the location recognition is feasible to optimize and can provide safety to considerably increased complexity of drone operations.

Definitions of on-board vision landing systems focused on landing on a known area rather than the strategies in unfamiliar and imprecise conditions. In that respect, Kong et al.[23] enounced the UAS landing on controlled zones as a well-studied research area by outlining the common use of motion algorithms and the requirement for structured landing settings. While fixed-wing UAS autonomous landing on a known area is commonly practiced on a runway,[24–26] drone and rotary-wing UAS studies concentrate on marker based tracking methods for landing.[23] Traditionally, a helipad of a known shape is used for target detection. Drones automatically updates landing target parameters based on visual markers and follow a path to the helipad. For instance, Saripalli et al.[27] provided the foundation for vision-based landing target detection on real-time which leads the act to a board behavior based controller to follow a path to a helipad of with a H-shaped marker. In order to achieve autonomy in such a scheme, on-board sensors might have needed to cope with UAS's processing power which was generally limited due to low weight capacity. To deal with this, Wenzel et al.[28] proposed a tracking approach that uses commodity consumer hardware. More recently, some researches appear to agree that image frames can be processed by a marker processing framework which includes various stages such as image rectification, conversion into a binary image and vision aided landmark recognition.[29,30] Similarly, Sudevan

et al.[31] fused speeded up robust features detector and fast approximate nearest neighbor method for landing on a stationary target. Additionally, Saavedra-Ruiz et al.[32] presented a monocular visual system, using a software-in-the-loop for autonomous landing on a predefined landing pad and Cabrera-Ponce and Martínez-Carranza[33] used a flag posed on a pole to locate the landing platform nearby.

Even though marker recognition procedures can perform automatic aircraft landing stationary target, further processes are needed to achieve accurate landing on a moving platform. Besides, advancing technologies made possible successive waves of new sensors: from basic signal providers to complex camera systems. To process these multi-sensor information, Yang et al.[34] came up with a practical framework in which data from various sensors of rotary-wing UAS is analyzed for reliable navigation information as the aircraft approaches to the landing deck on a moving marine vehicle. Considering multi-sensor information, more complex data need a systematical algorithm rather than the traditional data-processing methods used before. Correspondingly, to fuse data from multiple sensors with an attempt to provide the reliable information for navigation, Yang et al.[34] developed an extended Kalman filter using a series of measurements observed over time. This method can provide unknown variable estimations which are in general more precise than the ones with a single measurement alone.

Landing on mobile marine vehicle is also supported by additional findings of Venugopalan et al.[5] who proposed an autonomously control algorithm to land on a pad placed over an autonomous kayak, Polvara et al.[35] who aims to address the landing pad as the deck of a ship and Weaver et al.[36] who create a scaled-down model of ship landing of an UAS onto a mobile unmanned surface

vehicle. These studies had a more or less explicit perspective on harsh condition landing prevalent in the marine environments as a result of winds and currents causing the stationary landing target to rock and drift. In a like manner, autonomous landing on unknown vehicle positions is extended to the implementation of autonomous approach on a car moving.[37] Although the vehicles in which these studies are conducted are mobile and operate in unknown positions, the aircraft still navigates to a known deck or landing pad rather than unstructured and unknown harsh 3D surroundings. This gap is common not only for these schemes, but also the most of other methodologies using a known marker or shape.

In response to this gap, Johnson et al.[38] tackled the concept of soft-landing capability in unknown and hazardous terrain, arguing that it will allow exploration of previously inaccessible environments with strong scientific importance. Scherer et al.[39] also critiqued the inability of UAS to identify and verify landing zones and approach paths by using not only plane fitting but also various factors such as wind direction, terrain and skid interaction, rotor and tail clearance, and approach, abort and ground paths. While their algorithm could incorporate these factors to fulfill autonomous landing at unprepared sites, their framework and results were based on a full-scale helicopter which selects its own landing sites. For small-scale micro air vehicles, especially those capable of operating outside line-of-sight, findings by De Croon et al.[40] highlights the significance of optic-flow based slope estimation for relatively fast maneuvers.

The models up to here allowed successful estimation of landing location; however, extracting local and position-invariant features has a potential role in unknown environments. Nguyen et al.[41] found this to be true in their study which included a convolutional neural network to extract trained features from captured images. However, their algorithm was based on estimating a marker's location with visible light camera sensor rather than to process field images for emergency landing. A similar work based on deep reinforcement learning, a hierarchy of Deep Q-Network, is used for landmark detection by not only managing with low-resolution landmark images from a mounted camera and also providing higher performance than human pilots in some conditions.[42] A later work of image recognition technology in emergencies by Yang et al.[43] critiqued the drone navigation methods relying on global positioning system signals and introduced a landing procedure that can estimate drone's position by creating a grid map of the environment to decide on the most suitable landing location via a filtering algorithm. Rojas-Perez et al.[44] implemented CNNs for automatic detection zone for UAS in urban environments with a public dataset and synthetic data. In a further work, Osuna-Coutiño and Martinez-Carranza[45] extended the use of the CNN-based approach processing

a single image seeking to interpret areas where the human-made structures are observed. Lopez-Campos and Martinez-Carranza[46] advanced the synthetic data application by generating photogrammetric aerial-images from photo-realistic scenes.

It is increasingly evident that the innovations on image processing will continue to be a critical strategy for autonomous emergency landing. One can also expect that these vision-based interactions will also become a key criterion for supporting elements of safety and can collaborate with them. Recent works has begun to safety challenges and focused on third-party risk associated with UAS operations such as the people on the ground with no involvement in the operation.[47,48] Similar efforts were devoted by Lum and Waggoner[49] to remove the threats to human safety from mid-air collisions, as well as the ground strikes. To enable the tracking of fatality rates caused by crashes over time, Melnyk et al.[50] used historical data which gives key insights to enable operational safety in civilian airspace. While initial considerations relating to third party risks were around airports such as crash, individual risks, and societal concerns,[51] later studies tackled the risk management of unmanned flights over inhabited and populated areas.[52,53] Despite these works, little is known about the image processing for automated landing of larger drones (> 7 kg) in unstructured environments. Consequently, there is an incomplete picture of the way image characteristics are accurately analyzed under these conditions. That is why, this research attempts to identify key development stages of the image processing and ties them to safe landing and recovery of larger drones. Essentially, the research responds to the call for a novel approach about the information processing and distributed communication nodes in location recognition and draws inspiration from ever increasing regulations which have stressed the need for designing individually reliable drones that will enable end-users to ensure safe, sustainable, and secure operations.

Methodology

This section provides information on the proposed research method and CNN procedures for the application of image detection. The method is a series of layers which helps extracting the image features and respond to the final fully connected network for classification. The layers are described in the following subsections.

CNN architecture

The first layer of a CNN, also known as the input layer, consist of artificial input neurons that bring the initial imagery data into the network structure for the subsequent layers. These initial data were available from a database of 10,000's of images captured from the SafeEYE lab that

has been mounted, integrated, and flown in a number of test flights on a DJI Matrice 600 drone[18] (see Figure 2). There were a number of test flights that had been conducted with the SafeEYE, primarily to collect data and to test the machine vision approach for the detecting areas of interest for landing or a crash without any human injury or fatality. Most prominently, three campaigns with a total of approximately 12 hours of flight were conducted in two sites in Denmark, an emergency responder training facility at Rørdal, Aalborg and a military training compound Brikby consisting of about 30 empty houses, located at Oksbøl Barracks. Both sites are in Denmark. These areas were also used to accomplish the task of flight tests with Home Guard personnel acting as city residents. Having both rural and urban ground textures while legally being an urban area was the reason why these sites were selected. Besides, the military field had an airspace restriction zone, which allowed the drone to capture images up to an altitude of 150 m.

To perform in such a large-scale image processing, this study proposes the use of a feed-forward neural network type in which the neurons are able to reply to some surroundings in the coverage range. Original structure (the LeNet architecture - LeNet5 - 1990s) was first introduced by LeCun and pioneered the CNNs which propelled the field of Deep Learning.[11] The methodology uses a slightly modified version of the LeNet architecture and classifies the input images into two categories: "landing fields" and "not landing fields". The reason is that the structure is straightforward, relatively small in terms of memory footprint and could even run on a single-board computer, making it ideal for the use of SafeEYE lab.

A CNN usually receives an order 3 tensor input image with rows, columns, and channels which then sequentially proceed a series of processing steps, commonly known as

Figure 2. SafeEYE lab is mounted on a DJI M600 drone. The white box at the bottom is SafeEYE lab, with an extra IMU on top (orange). Behind SafeEYE lab is a standard X5 camera. The payload radio is mounted on the top of the aircraft. The image is from Oksbøl on December 4, 2019.

layers.[54] The abstract description of the CNN architecture can be given as:

$$x^1 \rightarrow w^1 \rightarrow x^2 \rightarrow w^2 \rightarrow \cdots \rightarrow x^{L-1} \rightarrow w^{L-1} \rightarrow x^L$$
$$\rightarrow w^L \rightarrow z \tag{1}$$

This processing goes through up to the point that all layers (L) are completed, which outputs z.[54] Accordingly, the in-flight frames from the SafeEYE lab are split into smaller ones with a resolution of $180 \times 180 \times 3$, for height, width and RBG. When the framework was applied to analyzing visual imagery, the layer formed of these frames served as a tensor with shape (images) $\times$ (height) $\times$ (width) $\times$ (depth). The output layer for training was chosen by visual inspection of input images and it determined green background like grass fields as a suitable landing site where there were limited obstacles on the frame. Such images had little or no variations such as different colors, shadows, rocks, trees, houses, tracks, etc. Between the input and output layer, there are multiple layers of the CNN: an initial convolution to polling followed by a second set on the same order, then an activation to a fully-connected layer, a second activation function to an another fully-connected layer and at the end, a soft max classifier (see Figure 3).

$$\text{In} \Rightarrow \text{Conv} \Rightarrow \text{Act} \Rightarrow \text{Pool} \Rightarrow \text{Conv} \Rightarrow \text{Act} \Rightarrow \text{Pool}$$
$$\Rightarrow \text{FC} \Rightarrow \text{Act} \Rightarrow \text{FC} \tag{2}$$

Convolution

The first building block in the framework is a convolution operation in which the feature detectors serves as CNN's filters. In this stage, a fairly simple 2D convolution operation begins with a kernel, a matrix of weights. Then, this slide over the input on 2D space, as seen on Figure 4 . This allows a element-wise matrix multiplication with the corresponding input section, and then summing up the matrix into a single production (Figure 4, dark square) which will be placed in another 2D feature matrix (the green grid).

In mathematical aspect, the convolution over a 2D input image "I" and 2D kernel "K" is given as $I * K$, denoting the operator with the symbol of $*$.[55]

$$G[m, n] = (I * K)[m, n]$$
$$= \sum_j \sum_k I[m - j, n - k]K[j, k], \tag{3}$$

where the result matrix indices are given by m and n for the rows and columns. All neurons in this layer were not only connected to a particular spot in the input volume, but to a full depth (see Figure 5).[56]
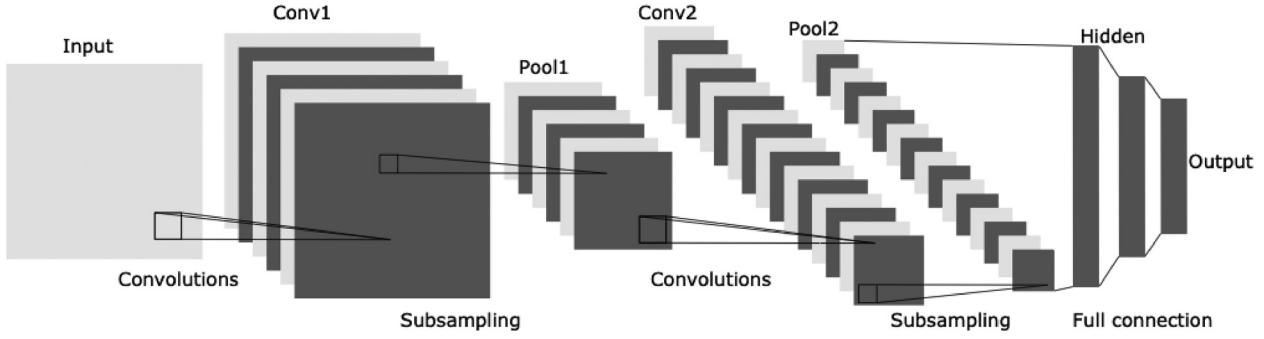
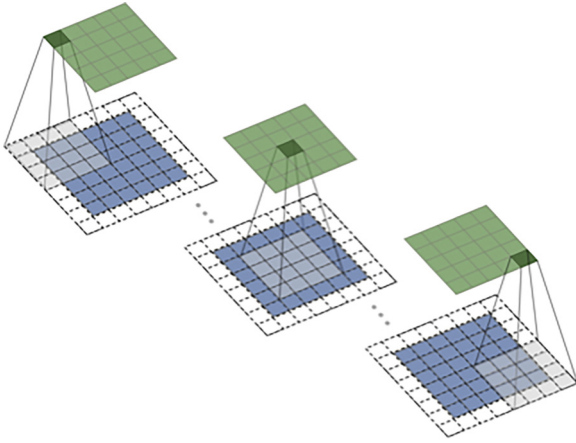**Figure 3.** Architecture of LeNet, image modified from LeCun et al.[11]

Figure 4. An example of the convolution operation: the blue grid is the input feature map where the kernel is the glared area.

Figure 5. A sample volume in the Convolutional layer.

Non Linearity - ReLU Layer

An additional operation called Rectified Linear Unit, or shortly ReLU, was added after every convolution step in the framework. ReLU, an essential unit of the CNN process, is a non-linear operation and its output is given as:

$$f(x) = x^+ = \max(0, \text{x}), \qquad (4)$$

This is applied per pixel and replaces all negative ones in the feature map by zero. ReLU is an outperforming activation functions comparing to others and widely acknowledged in the literature.[57]

Pooling - Sub Sampling

In the next part, a pooling layer (also called sub-sampling or down-sampling) is used for the dimensionality reduction of feature maps while retaining the most useful information. This layer progressively reduces the tensor size and network computation, and therefore it can control over-fitting.

The nexus in this research is 2D max pooling in which the largest element is taken from the spatial neighborhood as:

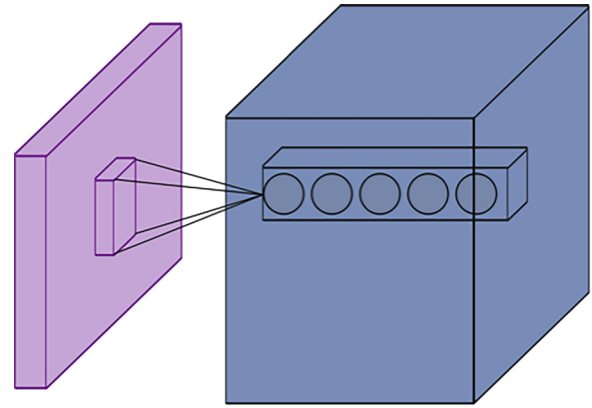$$f_{X,Y}(M) = \max_{i,j} M_{2X+i,2Y+j} \qquad (5)$$

where each operation is over 4 numbers with a stride of 2 sub-samples, discarding 75% of the content.

After executing the previous steps twice ("Conv $\Rightarrow$ Act $\Rightarrow$ Pool $\Rightarrow$ Conv $\Rightarrow$ Act $\Rightarrow$ Pool   "), the remaining pooled feature map is flattened into a single column and inserted into a ANN classification layer in the following step.

Fully Connected Layer- ANN Classification

Here is where the convolutional layers and a traditional Multi Layer Perceptron meet as the latter is included in the form of a "Fully Connected" layer. As the neurons from the previous layer are connected to all on the next layer, the framework at this point takes a more complex and advanced turn (see Figure 6). The basic unit in this computational model is the single-input neuron, also often called a node or unit, structure that is defined by definite functional operations of input (x), weight (w), bias (b) Lippmann[58]. To find the output, the neuron first takes the weighted sum of the input and weights and a bias term is added to this sum. The output is then passed through the activation function (f) to determine whether the neuron fires, and outputs a value.[55] The activation function
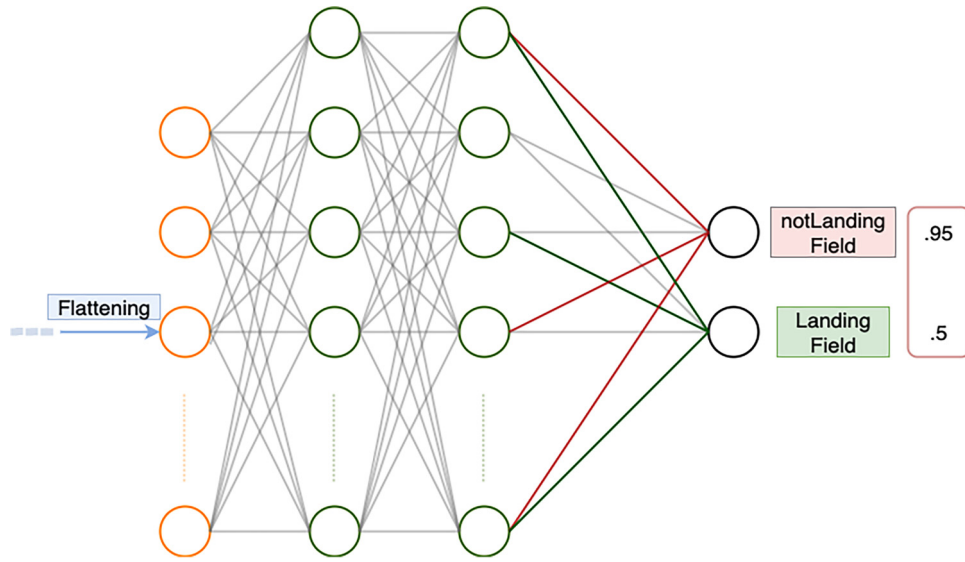
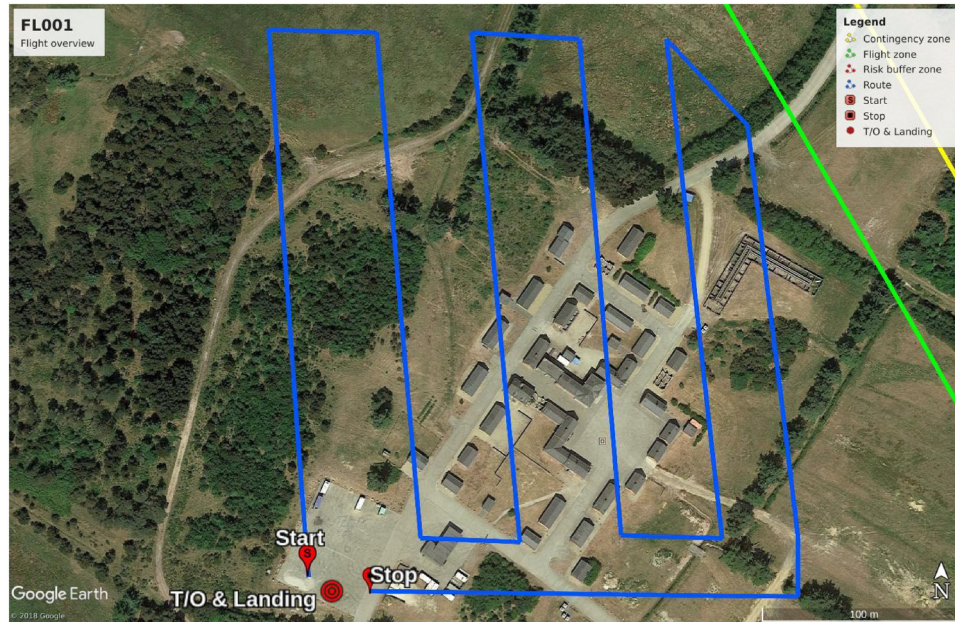**Figure 6.** Artificial Neural Network Classification Structure.

Figure 7. A flight plan at a military training area in Oksbøl. Some images from the flight is shown in Figure 8.

performs like a gateway that controls that the sum value is greater than a critical number. These are expressed as:

$$f(net), \quad \text{where net} = \sum_{i=1}^{n} w_i x_i + b \quad (6)$$

A fully connected neural network layer is formed by combining multiple of these units which simply associate input elements to an output, while in aggregate do complex computations. The multilayer perceptron arrange these units into a set of layers with some number of identical units so that it can learn non – linear functions. Units in a layer are connected to the next ones in the following layer, as seen in regular (non-convolutional) ANNs.

Softmax function

A softmax activation function is used in the output layer. This classifier is a generalization of the binary form of Logistic Regression and it takes a vector of arbitrary scores and normalizes it into a probability distribution.[59] In mathematical terms, the unit softmax function is defined as:

Figure 8. Ground images taken from the SafeEYE lab during the flight path shown in Figure 7. They are taken when the aircraft is following the right-most north-south leg.

Figure 9. Frames showing examples of (columns 1-3) *notlanding* locations, and (columns 4-6) *landing* locations.

$$\text{Softmax}(\mathbf{x})_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \qquad (7)$$

where the standard exponential function to each point x_i of the vector $\mathbf{x}$ is normalized into a vector of probabilities with a relative scale.

Case Study and Testing

The proposed image classification framework is applied with an up-close and detailed examination of autonomous drone landing in unknown environments. The goal is to provide a justification for the methods and to test whether the detection method is computationally

efficient enough to be installed on a small embedded computer.

Image acquisition

A large set of images have been collected with SafeEYE lab during a series of test flights. In all flights, SafeEYE lab (NanoPyH5 CPU: H5, Quad-core 64-bit high-performance Cortex A53 - a low powered system) mounted on a DJI M600 drone and flown at altitudes from 30 to 150 m over a variety of terrain. During flight, images are captured every few seconds with the SafeEYE lab camera and stored in raw format. All flight followed predetermine flight paths to ensure that the same ground features would be in images at varying altitudes, and to be able to reproduce the flights, if necessary. One example of a flight path is shown in Figure 7, and a couple of examples of the view from SafeEYE lab are shown in Figure 8.

The data collected from these campaigns provides sufficient ground images with various features from multiple attitudes and flight trajectories, vibration measures, and on-the-fly estimations.

All test flights were recorded on video and a few are made publicly available on the https://www.youtube.com/channel/UCwIUbrNZCwBuWZ4rRBUq3LAUAS-ability YouTube channel.[60] The videos starting with the date, i.e. "19.12.04", and flight number "FL00x" refers to the flight tests as mentioned above.

Table 1. Input data categories

Category	notLandingField	LandingField
Label	0	1
Count	9491	6786
Training Rate	75	75
Test Rate	25	25

Image Assessment

In Figure 8, sample images used for the classification algorithms are given. These examples are of a quality suitable for recognizing objects on ground and extracting the necessary features for image processing and classification. The imagery data set is a collection of such images taken from the camera mounted on SafeEye lab during the test flights. The raw images are then split up in smaller frames as shown in Figure 9. This is done to manually classify images according to the objects included in them and also to form an input data to the CNN structure. The frames has a resolution of $180 \times 180 \times 3$, for height, width and RBG as it was deemed as a sufficient size for a landing location. The frames are then be classified in two categories and labeled as *landing* and *notlanding* locations.

Both the output layer formation for training and the criteria for deciding whether a landing site is suitable are based on a visual inspection by a human expert. In general, the frames covered with limited color variations and green areas like grass fields are clustered as landing site whereas the ones with noise such as rocks, different colors, shadows, trees, houses, tracks etc. are categorized as not landing sites. Samples of both clusters are shown in Figure 9.

The goal of this data set is to classify areas of interest that can facilitate a landing or a crash without our human injury or fatality. The total number of images for landing evaluation is given in Tables 1 and tab:CNNLayerFormation. A common split of 75%/25% is used for partiting the data into training and test sets which are large enough to yield statistically meaningful results. Both sets are representative of data as a whole without having significantly different characteristics. The goal is to train the CNN model so that it can generalize well to new data. That is to say, the model does not over-fit the available data and it can do approximately as well on the test set as like the training set.

The overall training process of the methods provided in the previous section is summarized as follows:

Table 2. CNN Layer Formation

Level	Layer Type	Input Size	Convolution Kernel Size	Pooling Filter Size	Output Size
Input	Image - Input	–			28×28×1
1st Layer	Convolution	28×28×1	5×5		28×28×20
	Activation Function	28×28×20			28×28×20
2nd Layer	Pooling	28×28×20		2×2	14×14×20
3rd Layer	Convolution	14×14×20	5×5		14×14×50
	Activation Function	14×14×50			14×14×50
4th Layer	Pooling	14×14×50		2×2	7×7×50
5th Layer	Fully-Connected	7×7×50			500
	Activation Function	500			500
6th Layer	Fully-Connected	500			10
	Soft Max Classifier	10			10
Output	Classification - Output	10			2

- The network takes training images captured by SafeEYE camera as input.
- The filters and feature maps are initialized and applied to these images in the first convolutional layer.
- The network goes through the Rectified Linear Unit with an attempt to break up the image linearity.
- Then, the max pooling operation down-samples each feature map by calculating the largest value in each patch, so that it can highlight the most salient features and form pooled feature maps.
- A second set of convolutional layer is applied to the pooled feature maps.
- The second set of pooled feature maps are flattened and inserted into the artificial neural network classification function.
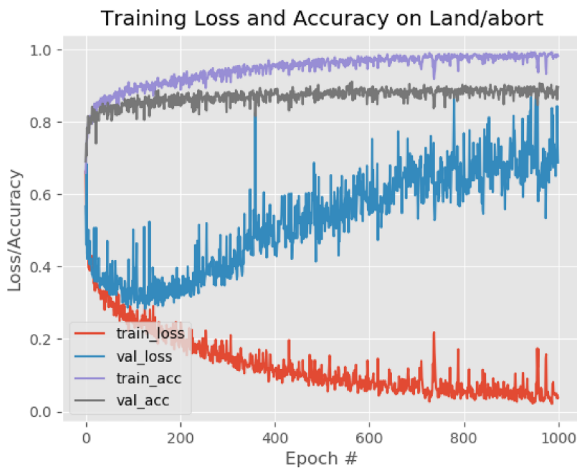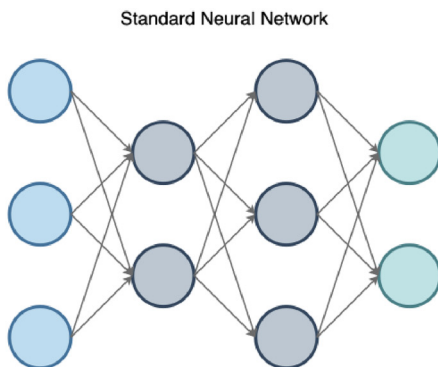- Softmax function is used as the last activation function to normalize the output.

These steps train the network with ground images. During this process, many iterations are required to update the network's parameters such as weights and feature maps so that the algorithm can reach an optimal performance point where the classification is accurate enough. To optimize these parameters, the Adam algorithm is implemented.[61] This is a stochastic gradient-based optimization method, based on adaptive estimations of lower-order moments.[61] The method allows straightforward implementation with little memory requirement and computational efficiency, and it is compatible with the complex cases that are large with regard to data and parameters.[61]

Results

The experimental findings in this section evaluate the imagery data collected for the work in the form of descriptive statistics. The section presents plots and graphs as well as the outcomes of relevant inferential statistical analyses. In the light of calculated scores and model performance, the structure is revised and altered. The results are reported in sufficient detail so that one can see what improvements

Figure 10. Illustration of the learning curves which are calculated by the metrics of the training loss (train_loss), training accuracy (train_acc), validation loss (val_loss), and validation accuracy over time (val_loss). These results are found with the initial settings introduced in the methodology section.
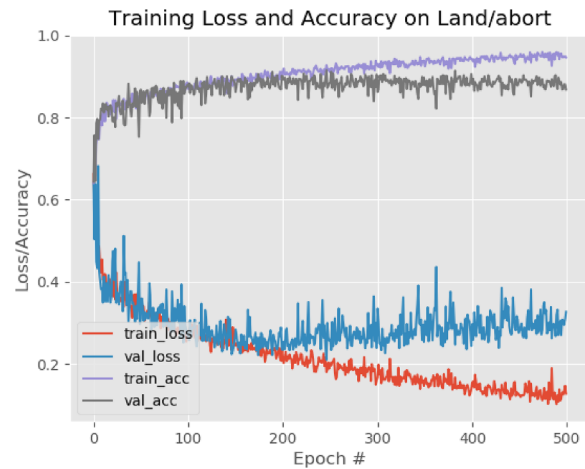
Figure 12. Results of the LeNet classifier with a dropout layer.

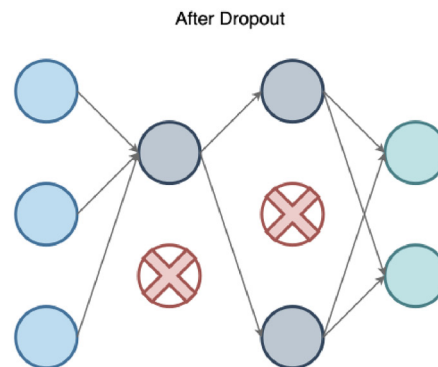**Figure 11.** An sample of how a thinned net production by applying dropout to a standard network

Figure 13. Results from the keras classifier after the dropout layer is applied. Top text shows whether the frame can be considered, and the percentage of confidence in the classification.
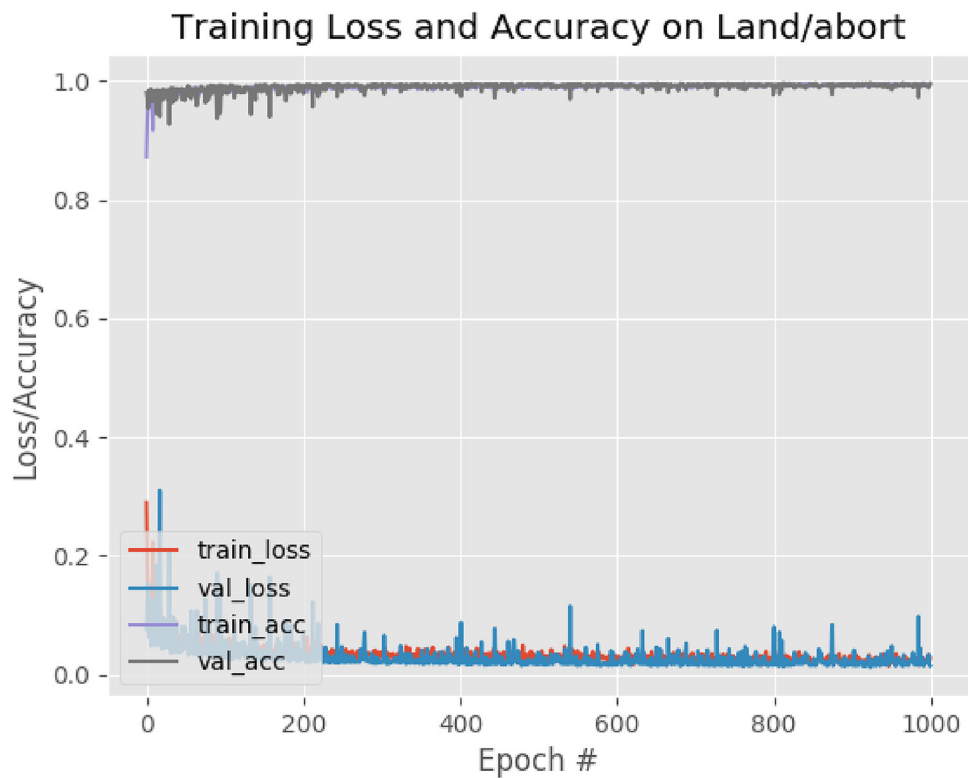

Figure 14. Refreshed learning curves of the classifier training after the data set is expanded

Figure 15. Results of the percentage of confidence in the classification from the image classifier after the model trained with the dropout layer and updated images from the new test flights.

were conducted and why, and to justify the proposed configurations. For processing times, the embedded computer (NanoPyH5 CPU: Allwinner H5, Quad-core 64-bit high-performance Cortex A53) could classify the flight snapshot (78 frames) into the two categories within 3 seconds.

Training CNN with the proposed settings

Results in this section demonstrates the applicability of the proposed default settings in the methodology section. Here, the study also compares the results of the initial data set with those of the further ground images. These results go beyond previous assumptions, showing that additional alterations, settings and data are necessary to provide an efficient ground image classification for autonomous landing.

The initial implementation of the model was with the image data set from the test flight at Rørdal in Aalborg, Denmark. A portion of the training data is separated into a validation data set to evaluate the model performance on this independent portion during each epoch.

The metrics of the training loss (train_loss), training accuracy (train_acc), validation loss (val_loss), and validation accuracy over time (val_loss) are used to judge the performance of the model. Accuracy calculates the percentage of the predictions that match with actual labels. The loss function is the "binary_crossentropy" which computes the cross-entropy loss between actual labels and predictions. The results of these metrics over time can be seen in Figure 10 in which the training part could fit the parameters of the model and produce improving results by iterations, but the validation part fails to provide an unbiased evaluation of a model fit while tuning parameters. Accordingly the following key findings emerge:

- The model is over-fitted, due to the gap between training and validation loss. The validation part cannot provide same decreasing results after the model has "passed" the training set so the training evaluation is biased to its data.
- Trained network corresponds too closely to the training images and makes an overly complex explanation the idiosyncrasies in the training data.

Figure 16. Further results from the classifier with a new cluster to display the classification accuracy

- The model over-memorizes the training samples and therefore fail to fit the test data
- The network cannot generalize to the test samples so that it cannot involve infer and apply the training findings to test split.

Thus, over-fitting becomes a critical problem in the proposed neural network with a large number of parameters and complex co-adaptations on training data. Even though it might be acceptable to have a gap between the training loss and validation loss curves, the validation loss should not be constantly increasing as witnessed in Figure 10. Using a dropout layer is an efficient way of addressing this problem Srivastava et al.[62]. The term refers to randomly "dropping out" units, both hidden and visible ones along with their connections,[62,63] from the neural network during training process.[62,63] Figure 11 illustrates how the units are temporarily removed from the network, along with incoming and outgoing links.

Accordingly, the proposed method can form a different architecture by randomly setting neuron input units to 0 with a frequency of rate at each step while others are scaled up by $1/(1 - \text{rate})$. Adding such a dropout layer allows a computationally inexpensive and effective regularization process along with a reduced over-fitting and an improved generalization error (see Figure 12). The classification accuracy increased as seen in the examples on Figure 13. The gap between training and validation loss was reduced and the classifier was able to correctly classify all the frames in Figure 13.

As seen by these classification results, the proposed method performed well at deciding on correct landing and non-landing locations. A further point to note is that the model with the dropout layer is a rather small model and could classify these initial frames in a reasonable time which is desired for the SafeEYE lab as it will not fill up the memory on the embedded computer.

Results with Additional Data

Even though there was a significant reduction in the gap between training and validation loss due to the new dropout layer placed between the two fully connected
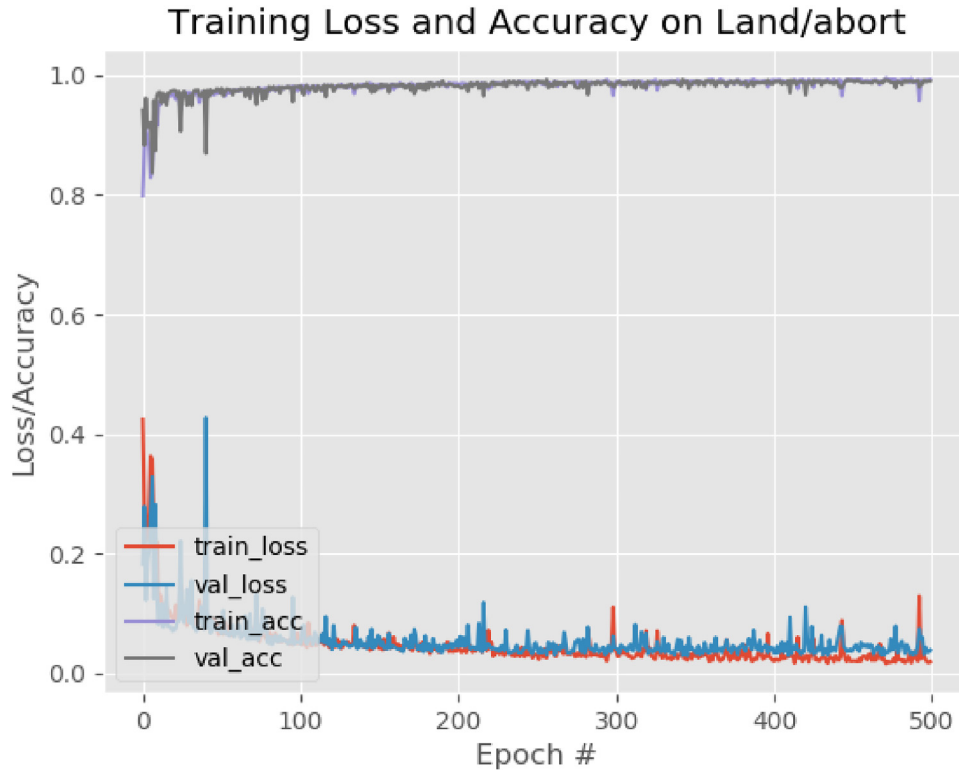
Training Loss and Accuracy on Land/abort

Figure 17. Illustration of the learning curves on the orthophotos data set.

layers at the end of the neural network, the gap between them was still present after around 200 epochs.

The results were still not desirable but hinted two things. First, the data scarcity could have been a major bottleneck for the image recognition model to reach desired classification levels after around 200 epochs. Second, the model performance relied heavily on the size of available data and SafeEYE required more test flights. After the database was updated with new images from two additional flight tests, see section , the network training was re-run with an identical setup and the results were refreshed.

The results of the later experiment found clear support for both using a dropout layer and increasing the data source. Figure 14 illustrates the performance of classification which delivered significantly better results than Figure 10 due to the proposed alterations. This yielded increasingly well assortment on both landing and non-landing fields because both training and validation loss did not deviate from each other as the previous models. The reason for this was understandably due to the diversity of data and higher variations in frames.

The frames in Figure 15 illustrate the overall results from the classification of frames from the military training compound. The model performs well at selecting frames where it is not suitable to land. The promising finding was that the framework could detect the noise in the frames accurately and could classify the notlandingField frames when there

are any form of objects, structures, trees or natural obstacles at the image.

The applicability of these new results can also be seen on the for landing locations on Figure 15. The model gave clearly well results to correctly find the grass fields as landing location, even if there were sometimes lower classification score as seen in the top middle frame. However, this could be due to the fact that the image was slightly distorted from the fish eye effect of the camera, or alternatively it was due to the slight discoloration in the ground texture.

Despite the general success for the classification, inconsistent results might be obtained for certain cases. Areas with small objects or variations had edges associated with them. Safer fields, on the other hand, were free from these edges. However, some frames with indistinct edges posed some problems when carrying out the classification. To highlight these frames, the results of classification accuracy were split into three different legend colors: green for > 70% LandingField classification accuracy, blue for < 70% LandingField classification accuracy and red for NotLandingField.

As seen on Figure 16, the model was able to correctly classify the first row where the drone can land and the second row where it cannot. However, the third row is more challenging, especially for the last frame which seems a safe place to land but labelled as otherwise. This might be a minor drawback of the variance in the data

Figure 18. Results of the image classifier on the orthophotos frames.

present in the real world. In the convolutional layer, the model loses some information about the frame composition, and also position, and transmits incomplete knowledge further layers which might not be able to classify correctly. When the objects are under different angles, backgrounds, or lighting status, the model may not find accurate features and other information signs to classify the frame as intended. This might be the reason why the frame on the right corner of Figure 16 provided misclassification. On the other hand, the white variation could be rocks or any other obstacles in landscape level and it would be hard to say whether the model fails in classification in this example. Additionally, the convolutional networks could successfully recognize the landing images in terms of autonomous landing safety.

Summary of findings

Imagery differences can occur in different forms and be observed in interaction with various factors in different domains. To validate whether the framework could perform in such cases, an alternative data set of ground images created from orthophotos[64] is also used as a benchmark to measure how the model performs with other images collected at different settings. This can confirm the reliability of the results by a comparison between the control frames and the original SafeEYE recordings. SDFE[64] provides these open aerial photograph data which are orthorectified and geometrically corrected such that the scale is uniform. The data follow a given map projection. Like the aerial photographs taken during a drone flight, the orthophotos can be used in image analysis tasks in emergency drone landing, since they are accurate representation of unknown enviromental surfaces. The results of the model trained on the orthophotos (see Figure 17) lead to similar conclusion where there is a low training and validation loss, and a high training and validation accuracy.

As before, it was able to correctly classify the locations as can be seen on the first and second rows of Figure 18. The frames on the bottom row, on the other hand, involve different objects that cannot be found on the drone captured data set such as unpaved road surface and small geological formations on the field. The model had the capability to rank these frames with a lower score and it proved that it

does not lack the consideration of spatial relationships while processing the ground images. As the primary goal in an emergency situation is to land (or crash in the worst case) of the drone in a desired location, such ranking can be of use while selecting the best frame location.

Conclusion

This work investigated the potential role of convolutional neural network based image classification for drone emergency landing. It is an original study that demonstrates the practicality of vision-based location detection on the unknown ground. CNN model allowed to encode certain image properties into the network architecture so that the forward function was more efficient to implement with reduced number of parameters. The results clearly revealed that the model was able to successfully classify landing environments and suggest relevant captions. This is in line with contemporary drone studies in the autonomous flight context postulating that artificial intelligence can perceive drone environment and take related actions for safer operations. Some limitations might be related to splitting the drone captured images into frames and applying the model to these frames. Although the classification performance was deemed acceptable, a planning through a dynamic environment might have a further positive effect on landing location classification. The confidence in the results can be strengthened with semantic-segmentation or visual scene-understanding algorithms that can operate in real-time on low-power drones. It would then be possible to make efficient use of scarce ground imagery available on embedded drone systems, compared to fully fledged workstations.

ORCID iD

Oghuz Bektash ![ORCID] https://orcid.org/0000-0003-3687-3703

References

1. Wu P and Clothier R. The development of ground impact models for the analysis of the risks associated with unmanned aircraft operations over inhabited areas. In: *Proceedings of the 11th International Probabilistic Safety Assessment and Management Conference and the 2012 Annual European Safety and Reliability Conference*, pp.5222–5235. Curran Associates/International Association of Probabilistic Safety Assessment and Management, 2012.

2. EU. Commission Implementing Regulation (EU) 2019/947 of 24 May 2019 on the rules and procedures for the operation of unmanned aircraft (Text with EEA relevance.), *Official Journal of the European Union*, vol. 2019, May 2019.

3. EU. Commission Delegated Regulation (EU) 2019/945 of 12 March 2019 on unmanned aircraft systems and on third-country operators of unmanned aircraft systems, *Official Journal of the European Union*, vol. 2019, March 2019.

4. Vidal VF, Honório LM, Santos MF et al. UAV vision aided positioning system for location and landing. In *2017 18th international carpathian control conference (ICCC)*, IEEE, May 2017, pp.228–233.

5. Venugopalan TK, Taher T and Barbastathis G. Autonomous landing of an Unmanned Aerial Vehicle on an autonomous marine vehicle. In: *2012 Oceans*, IEEE, October 2012, pp.1–9.

6. Zhao T and Jiang H. Landing system for AR. Drone 2.0 using onboard camera and ROS. In: *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, IEEE, August 2016, pp.1098–1102.

7. Skoczylas M. Vision analysis system for autonomous landing of micro drone. *acta mechanica et automatica* 2014; 8: 199–203. https://doi.org/10.2478/ama-2014-0036

8. Kim J, Lee YS, Han SS et al. Autonomous flight system using marker recognition on drone. In: *2015 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, IEEE, January 2015, pp.1–4.

9. Torres RN and Fraternali P. Learning to identify illegal landfills through scene classification in aerial images. *Remote Sens (Basel)* 2021; 13: 4520.

10. Akbari Y, Almaadeed N, Al-maadeed S et al. Applications, databases and open computer vision research from drone videos and images: a survey. *Artif Intell Rev* 2021; 54: 3887–3938.

11. LeCun Y, Bottou L, Bengio Y et al. Gradient-based learning applied to document recognition. *Proc IEEE* 1998; 86: 2278–2324.

12. Krizhevsky A. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

13. Zeiler MD and Fergus R. Visualizing and understanding convolutional networks. In: *European conference on computer vision*, Springer, Cham, September 2014, pp.818–833.

14. Simonyan K and Zisserman A. Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199.*, 2014.

15. Simonyan K and Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556.*, 2014.

16. Carnie R, Walker R and Corke P. (2006, May). Image processing algorithms for UAV" sense and avoid". In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*, IEEE, pp.2848–2853.

17. Susini A. A technocritical review of drones crash risk probabilistic consequences and its societal acceptance. *Lnis* 2015; 7: 27–38.

18. Bektash O, Pedersen JN, Gomez AR et al. Automated Emergency Landing System for Drones: SafeEYE Project. In: *2020 International Conference on Unmanned Aircraft*

Systems (ICUAS), IEEE, September 2020, pp.1056–1064. doi:10.1109/ICUAS48674.2020.9214073.

19. Bektash O and la Cour-Harbo A. Vibration analysis for anomaly detection in unmanned aircraft. *Annu Conference Prognostics Health manag Soc* 2020; 12: 10. DOI:10.36001/phmconf.2020.v12i1.1143SEP.

20. Skoczylas M, Rakowski W, Cherubini R et al. Unstained viable cell recognition in phase-contrast microscopy. *Opto-Electron Rev* 2011; 19: 307–319.

21. Ding D, Yoon J and Lee C. Traffic sign detection and identification using SURF algorithm and GPGPU. In: 2012 International SoC Design Conference (ISOCC), IEEE, November 2012, pp.506–508.

22. Pan J, Chen W and Peng W. A new moving objects detection method based on improved SURF algorithm. In *2013 25th Chinese Control and Decision Conference (CCDC)*, IEEE, May 2013, pp.901–906.

23. Kong W, Zhou D, Zhang D et al. Vision-based autonomous landing system for unmanned aerial vehicle: A survey. In: *2014 international conference on multisensor fusion and information integration for intelligent systems (MFI)*, IEEE, September 2014, pp.1–8.

24. Ettinger SM, Nechyba MC, Ifju PG et al. Vision-guided flight stability and control for micro air vehicles. *Adv Robot* 2003; 17: 617–640.

25. Gonçalves TF, Azinheira JR and Rives P. Vision-based automatic approach and landing of fixed-wing aircraft using a dense visual tracking. In: *Informatics in Control Automation and Robotics*, Springer, Berlin, Heidelberg, 2011, pp.269–282.

26. Laiacker M, Kondak K, Schwarzbach M et al. Vision aided automatic landing system for fixed wing UAV. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, November 2013, pp.2971–2976.

27. Saripalli S, Montgomery JF and Sukhatme GS. Vision-based autonomous landing of an unmanned aerial vehicle. In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, IEEE, May 2002, Vol. 3, pp.2799–2804.

28. Wenzel KE, Rosset P and Zell A. Low-cost visual tracking of a landing place and hovering flight control with a microcontroller. *J Intell Robot Syst* 2010; 57: 297.

29. Vidal VF, Honório LM, Santos MF et al. UAV vision aided positioning system for location and landing. In: *2017 18th international carpathian control conference (ICCC)*, IEEE, May 2017, pp.228–233.

30. Zhao T and Jiang H. Landing system for AR. Drone 2.0 using onboard camera and ROS. In *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, IEEE, August 2016, pp.1098–1102.

31. Sudevan V, Shukla A and Karki H. Vision based autonomous landing of an Unmanned Aerial Vehicle on a stationary target. In: *2017 17th International Conference on Control, Automation and Systems (ICCAS)*, IEEE, October 2017, pp.362–367.

32. Saavedra-Ruiz M, Pinto-Vargas AM and Romero-Cano V. Monocular visual autonomous landing system for quadcopter drones using software in the loop. *IEEE Aerospace and Electronic Systems Magazine*, 2021.

33. Cabrera-Ponce AA and Martínez-Carranza J. Onboard CNN-Based Processing for Target Detection and Autonomous Landing for MAVs. In: *Mexican Conference on Pattern Recognition*, Springer, Cham, June 2020, pp.195–208.

34. Yang X, Mejias L and Garratt M. Multi-sensor data fusion for UAV navigation during landing operations. In: *Proceedings of the 2011 Australian Conference on Robotics and Automation*. Australian Robotics and Automation Association Inc., Monash University, 2011, pp.1–10.

35. Polvara R, Sharma S, Wan J et al. Towards autonomous landing on a moving vessel through fiducial markers. In: *2017 European Conference on Mobile Robots (ECMR)*, IEEE, September 2017, pp.1–6.

36. Weaver JN, Frank DZ, Schwartz EM et al. UAV performing autonomous landing on USV utilizing the robot operating system. In: *Proc. of the ASME District F-Early Career Technical Conference, Citeseer*, November 2013.

37. Baca T, Stepan P and Saska M. Autonomous landing on a moving car with unmanned aerial vehicle. In: *2017 European Conference on Mobile Robots (ECMR)*, IEEE, September 2017, pp.1–6.

38. Johnson A, Montgomery J and Matthies L. Vision guided landing of an autonomous helicopter in hazardous terrain. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, IEEE, April 2005, pp.3966–3971.

39. Scherer S, Chamberlain L and Singh S. Autonomous landing at unprepared sites by a full-scale helicopter. *Rob Auton Syst* 2012; 60: 1545–1562.

40. De Croon GCHE, Ho HW, De Wagter C et al. Optic-flow based slope estimation for autonomous landing. *Int J Micro Air Vehicles* 2013; 5: 287–297.

41. Nguyen PH, Arsalan M, Koo JH et al. LightDenseYOLO: A fast and accurate marker tracker for autonomous UAV landing by visible light camera sensor on drone. *Sensors* 2018; 18: 1703.

42. Polvara R, Patacchiola M, Sharma S et al. Toward end-to-end control for UAV autonomous landing via deep reinforcement learning. In: *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, June 2018, pp.115–123.

43. Yang T, Li P, Zhang H et al. Monocular vision SLAM-based UAV autonomous landing in emergencies and unknown environments. *Electronics* 2018; 7: 73.

44. Rojas-Perez LO, Munguia-Silva R and Martinez-Carranza J. Real-time landing zone detection for UAVs using single aerial images. In: *10th International Micro Air Vehicle Competition and Conference, Melbourne, Australia*, November 2018, pp.243–248.

45. Osuna-Coutiño JDJ and Martinez-Carranza J. Structure extraction in urbanized aerial images from a single view using a CNN-based approach. *Int J Remote Sens* 2020; 41: 8256–8280.

46. Lopez-Campos R and Martinez-Carranza J. ESPADA: Extended synthetic and photogrammetric aerial-image dataset. *IEEE Robot Autom Lett* 2021; 6: 7981–7988.

47. Ancel E, Capristan FM, Foster JV et al. Real-time risk assessment framework for unmanned aircraft system (UAS) traffic management (UTM). In: *17th aiaa aviation technology, integration, and operations conference*, 2017, p.3273.

48. Bertrand S, Raballand N, Viguier F et al. Ground risk assessment for long-range inspection missions of railways by UAVs. In: *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2017, pp.1343–1351.

49. Lum C and Waggoner B. A risk based paradigm and model for unmanned aerial systems in the national airspace. In: *Infotech@ Aerospace 2011*, 2011, p.1424.

50. Melnyk R, Schrage D, Volovoi V et al. A third-party casualty risk model for unmanned aircraft system operations. *Reliab Eng Syst Saf* 2014; 124: 105–116.

51. Evans A, Foot PB, Mason SM et al. Third party risk near airports and public safety zone policy. *R&D Report 9636*, 1997.

52. Clothier R, Walker R, Fulton N et al. A casualty risk analysis for unmanned aerial system (UAS) operations over inhabited areas. In: *Proceedings of AIAC12: 2nd Australasian Unmanned Air Vehicles Conference*, Bristol UAV Conference, 2007, pp.1–16.

53. Burke DA, Hall Jr CE and Cook SP. System-level airworthiness tool. *J Aircr* 2011; 48: 777–785.

54. Wu J. Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology Nanjing University China* 2017; 5: 23.

55. Rosebrock A. Deep learning for computer vision with python: Starter bundle. *PyImageSearch* 2017: 330.

56. Karpathy A. Cs231n convolutional neural networks for visual recognition. *Neural Netw* 2016; 1.

57. LeCun Y, Bengio Y and Hinton G. Deep learning. *Nature* 2015, May; 521: 436. DOI:10.1038/nature14539.

58. Lippmann RP. An introduction to computing with neural nets. Artificial neural networks. *Theor Concept* 1988; 209: 36–54.

59. Goodfellow I, Bengio Y and Courville A. *Deep learning*. Cambridge, MA: MIT Press, 2016.

60. UAS-ability. Home, [youtube channel]. n.d., https://www.youtube.com/channel/UCwIUbrNZCwBuWZ4rRBUq3LA (accessed 25 August 2020).

61. Kingma DP and Ba J. Adam: A method for stochastic optimization. *arXiv preprint arXiv*:1412.6980, 2014.

62. Srivastava N, Hinton G, Krizhevsky A et al. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 2014; 15: 1929–1958.

63. Hinton GE, Srivastava N, Krizhevsky A et al. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv*:1207.0580, 2012.

64. SDFE - The Danish Agency for Data Supply and Efficiency. *About the Danish Map Supply*. https://kortforsyningen.dk/indhold/english (2016, accessed 06 September 2020).