Deep Learning Based Obstacle Awareness from Airborne Optical Sensors

Manogna Ammalladene-Venkata* Omkar Halbe[1] Christian Seidel
Avionics and Systems Engineering
Airbus Helicopters Deutschland GmbH
Donauwörth, Germany

Professor of Intelligent Autonomous Flight Guidance
Technische Hochschule Ingolstadt
Ingolstadt, Germany

Christine Groitl Lothar Kramel Christoph Stahl Heiko Seidel
Intelligence Surveillance Reconnaissance Systems Engineering
Airbus Defense & Space GmbH, Manching, Germany

Aviation statistics identify collision with terrain and obstacles as a leading cause of helicopter accidents. Assisting helicopter pilots in detecting the presence of obstacles can greatly mitigate the risk of collisions. However, only a limited number of helicopters in operation have an installed helicopter terrain awareness and warning system (HTAWS), while the cost of active obstacle warning systems remains prohibitive for many civil operators. In this work, we apply machine learning to automate obstacle detection and classification in combination with commercially available airborne optical sensors. While numerous techniques for learning-based object detection have been published in the literature, many of them are data and computation intensive. Our approach seeks to balance the detection and classification accuracy of the method with the size of the training data required and the runtime. Specifically, our approach combines the invariant feature extraction ability of pretrained deep convolutional neural networks (CNNs) and the high-speed training and classification ability of a novel, proprietary frequency-domain support vector machine (SVM) method. We describe our experimental setup comprising the CNN+SVM model and datasets of predefined classes of obstacles—pylons, chimneys, antennas, TV towers, wind turbines, helicopters—synthesized from prerecorded airborne video sequences of low-altitude helicopter flight. We analyze the detection performance using average precision, average recall, and runtime performance metrics on representative test data. Finally, we present a simple architecture for real-time, onboard implementation and discuss the obstacle detection performance of recently concluded flight tests.

Introduction

Controlled flight into terrain and collision with obstacles continue to be the leading causes of helicopter accidents (Refs. 1, 2). Pilot judgment and pilot situational awareness have overwhelmingly contributed to such accidents according to these studies. Accident statistics are particularly high for helicopters that operate in low-altitude missions and without any terrain and obstacle warning equipment.

Airbus Helicopters is committed to ensuring a high-safety standard for its fleet and operators. The company has put in place measures across the complete life cycle, from design, engineering, and production to maintenance, training, and partnerships towards an ambitious target for aviation safety. (See https://www.airbus.com for details on Airbus Helicopters initiatives for aviation safety). Knowing that accidents caused by collision with terrain and obstacles represent a large proportion of the overall accidents, offering operators with cost-effective obstacle awareness systems is a logical step towards enhancing operational safety.

Contemporary solutions for obstacle awareness include helicopter terrain awareness and warning systems (HTAWS) and active sensor-based awareness systems. HTAWS produces cautions and alerts based on predictive algorithms that project the helicopter flight path and assesses terrain and obstacle conflicts using an on-board database (Ref. 3). The terrain and obstacle databases on which the alerting decisions are based are issued by aeronautical information publications. However, many of the obstacle databases are overwhelmingly catered to the needs of commercial fixed-wing operations so that obstacle data are rich in the vicinity of airports but often poor in accuracy, coverage, and exhaustiveness elsewhere. As helicopter missions become more complex and new rotorcraft operations evolve in increasingly cluttered environments, real-time, onboard, sensor-based obstacle detection capability will be a key enabler to assist helicopter pilots in preventing mishaps.

Prior Work on Obstacle Sensing

Prior work on obstacle sensing has almost exclusively focused on radar and LiDAR (light detection and ranging) sensors, a review of which is given in Ref. 4. Airbus Helicopters has extensively evaluated and flight-tested novel active and passive sensing technologies for rotor strike alerting (Ref. 5), flight in degraded visual environments (DVE) (Ref. 6), and automatic offshore rig approaches (Ref. 7). Military operators have also benefitted from advanced LiDAR sensors to mitigate the risk of collisions during low-level flights in harsh environments. These include the obstacle warning system (OWS) installation on the NH-90 helicopter (Ref. 8) that was also tested on the UH-60 helicopter (Ref. 9). Recently, improved versions of the OWS were tested on the German Aerospace Center (DLR) experimental EC135 helicopter (Ref. 10) and an Airbus experimental H145 helicopter (Ref. 6). Recent U.S. Army research into helicopter mission autonomy also used LiDAR sensors for obstacle field navigation and safe landing area determination (Ref. 11).

*Corresponding author; email: manogna.ammalladenevenkata@gmail.com.
Presented at the Vertical Flight Society's 77th Annual Forum & Technology Display, Virtual, May 10–14, 2021. Winner of the Best Paper Award in Avionics & Systems and Best Overall in Systems Integration disciplines. Manuscript received January 2022; accepted October 2022.
[1]Presently at the CSIR–National Aerospace Laboratories, Flight Mechanics and Control Division, Bangalore, India.

Although the aforementioned active sensing technologies based on light and/or radio waves are promising, many civilian operators and missions demand equivalent system performance at low SWaP-C (size, weight, power, cost). These requirements continue to pose challenges to helicopter manufacturers and prevent large-scale commercial exploitation of active obstacle awareness systems. Since many civilian helicopters continue to fly mostly in visual meteorological conditions (VMC), and despite a considerable risk of encountering marginal VMC or inadvertent entry into instrument meteorological conditions (IMC) imaging sensors in the visible spectrum paired with advanced image processing algorithms can be a promising start for situational awareness enhancements and piloting aids.

Computer Vision-Based Obstacle Detection: Opportunities and Challenges

Towards improving pilot situational awareness at acceptable SWaP-C, the detection and classification of potential threats and obstacles by means of commercially available, airborne optical sensors holds great promise. Recent advances in data-driven computing have shown the potential to automate detection, recognition, and identification tasks in vision systems. By combining camera images from different spectral bands, a well-trained deep neural network architecture model hosted on embedded graphical processing hardware can accurately detect and localize obstacles in the sensor field of view. The detected obstacles can then be used to either augment HTAWS obstacle databases, improve the performance of active sensors, or simply be shown to the pilot to draw attention to a potential threat. However, developing a reliable computer vision-based obstacle detection system involves many technical challenges. First, training deep neural networks typically requires large datasets, whereas aerial images of obstacles for training purposes are typically very limited and expensive. Second, both training and implementation of deep neural networks are computationally expensive tasks. Therefore, the solution needs to be robust to the limited datasets and yet achieve acceptable runtime performance on embedded airborne computers.

To address these challenges, we propose an efficient obstacle detection and classification technique using monocular two-dimensional images from the helicopter-fixed forward-looking camera and evaluate its performance on desktop and miniaturized graphical processing platforms. Specifically, our approach combines pretrained, highly optimized deep convolutional neural networks (CNN) feature extractors and a novel, proprietary, high-speed frequency-domain support vector machine (SVM) implementation recently patented in Ref. 12. In this work, the term "detection" refers to the identification of the presence of obstacles and their localization in the image coordinates, whereas the term "classification" refers to the identification of the class/type of the detected obstacles. We, however, do not treat the subject of obstacle localization in the global (world) coordinates in this work. Likewise, we do not address compliance demonstration and product certification aspects of the onboard system installation.

The objectives of our work towards a concept for vision-based obstacle awareness can be summarized as follows:

1) define the operational scope and requirements for obstacle awareness;

2) gather representative datasets for training purposes;

3) train high-speed SVM classifiers and CNN feature encoders for real-time obstacle detection and classification in forward-looking optical cameras;

4) evaluate detection and runtime performance on a desktop workstation and an airborne embedded general-purpose graphical processing unit (GP-GPU).

The remainder of the paper is organized as follows. We first discuss the basic ideas of machine learning applied to computer vision, introduce the current state-of-the-art in artificial and convolutional neural networks (ANN, CNN) and SVM, and present some recent automatic object detection and classification techniques. We discuss the merits and weaknesses of these techniques and propose our computer vision method that leverages the strengths while mitigating the drawbacks of these techniques for the automatic obstacle detection and classification problem. We then present our experimental setup for automatic detection and classification of obstacles from airborne optical sensors along with some basic performance metrics computed on a desktop computer. Finally, we present obstacle detection and classification results of the airborne installation consisting of an NVIDIA-embedded GP-GPU hosting our algorithm and a helicopter-enhanced vision system (HeliEVS) sensor on an experimental Airbus H145 helicopter.

Machine Learning Framework for Automatic Object Detection/Classification

Automatic object detection/classification has grown to be one of the most popular applications of computer vision and deep learning. It addresses the problem of identifying the presence of objects of interest in an image, their localization in the image coordinates, and their classification by types, all without human intervention. Typically, the task of object detection and classification can be divided into three main processes (Ref. 13):

1) division of the input image into informative regions;

2) extraction of features from each region;

3) classification and localization.

Objects of interest can be present at different pixel positions in the image in various sizes and aspect ratios. A natural solution to find all the possible regions of interest (RoI) is to scan the entire image using the sliding window technique. However, this generates a large number of windows, most of which may be redundant, making the process computationally expensive. On the contrary, if only a limited number of windows are considered, then some objects may be missed. Each detected object can be identified by its unique features using popular computer vision algorithms such as Scale Invariant Feature Transform (SIFT), Histograms of Oriented Gradients (HOG), Haar wavelets etc., to represent the object's features (Refs. 14, 15).

The next step is to classify the objects according to the desired object classes by recognizing their unique features. Some of the commonly used classification techniques are AdaBoost, SVM, and deformable part-based model (Ref. 15). Before the advent of modern neural networks, all these three steps were programmed manually and were designed individually based on the type of the data used. Recently, thanks to the ImageNet CNN architectures (Ref. 17), feature extraction and classification have been automated without the need for manual design of feature representors. In addition, object detection models like You Only Look Once (YOLO; Ref. 18), single shot detector (SSD; Ref. 19), and hybrid models (Ref. 20) also localize the objects by drawing bounding boxes around them using techniques like regression (Refs. 14, 15). These algorithmic evolutions have been accompanied by great strides in hardware performance in terms of GP-GPU. Figure 1 shows the structure of an object detection model from Ref. 16 including the different steps just discussed.

Convolutional neural networks

Artificial neural networks (ANNs) are inspired by the functions of biological brains. ANNs imitate the neuronal structure of the cerebral cortex of the human brain in a simplified way, which results in linear and nonlinear relationships between the input and output information

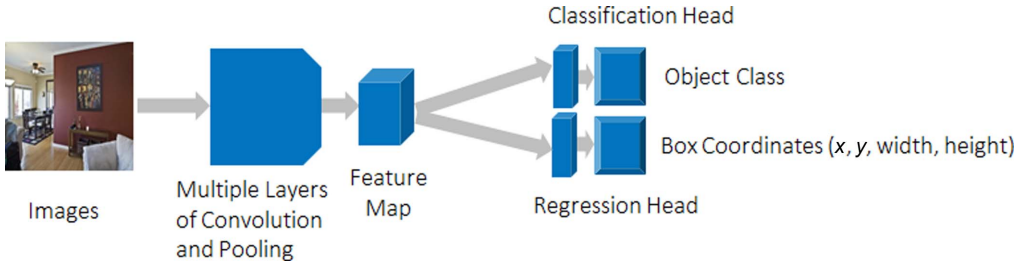

Fig. 1. A model of automatic object detection using deep learning (Ref. 16).

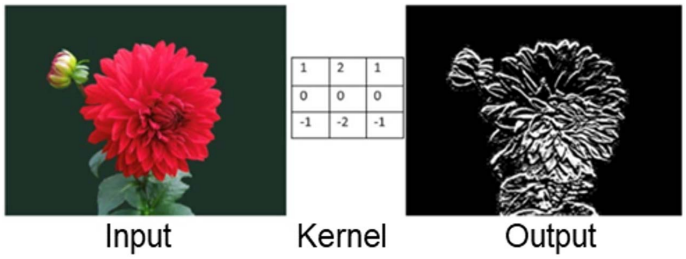

Fig. 2. Example of convolution operation in CNNs to extract vertical edges (Ref. 25).

consistently. ANNs can be trained with a large collection of data relevant to the task at hand. When the model is applied on new data, ANNs make appropriate inferences based on the patterns the model was trained to detect. Due to their widespread utility, ANNs have found applications in autonomous driving, pattern recognition, data analytics, medical data processing, and stock market prediction, among others (Ref. 21).

The term "deep learning" refers to ANNs that consist of multiple layers of artificial neurons in contrast to shallower network architectures that were predominantly used in the past. CNNs are a subdomain of deep learning that represents the current state of the art (Refs. 17, 22, 23, 24). CNNs have a similar structure as ANNs and are composed of a sequence of layers performing various operations to hierarchically learn and represent the data. The major difference is that CNNs perform the convolution operation on the input data with multiple successive kernels. A kernel is a filter that extracts features of interest from the input images. Mathematically, a kernel is a matrix that performs a dot product on functions of the input images. The elements of these kernels are learned during the training process. Also, unlike ANNs, the neurons in the preceding layer of CNNs are not linked to all the neurons in the successive layer but only to a small region around it. The size of the region corresponds to the size of the kernel or filter. An example is shown in Fig. 2, which illustrates the convolution operation on an input image to extract its vertical edges. The output image is the result of the convolution operation between the input image and the filter or kernel (Refs. 24, 25).

After learning the elements of all the kernels, the actual classification or regression task is performed by applying one or multiple "fully connected layers" as they are known from standard ANNs. This process leads to a significant reduction in parameters that need to be tuned and stored and thereby greatly improves the runtime and generalization performance of the network.

CNNs are most commonly used to solve classification problems. A CNN that does classification has two major components: feature extraction by the "hidden layers" and classification by the "fully connected layers." Feature extraction by the hidden layers involves a series of convolution and pooling operations to detect the relevant features from the input data, as seen in Fig. 1. Classification by the fully connected layers involves computing the probability scores and depth values of the given input data (Ref. 26).

Applying deep learning to computer vision tasks such as real-time automatic object detection/classification on high-definition video sequences remains a challenge. The best methods in this direction currently achieve a performance of about 5 frames/second for an image resolution that is much smaller than full-high-definition (full-HD) (Ref. 22).

Support vector machines

SVM is a classical machine learning technique popularly applied to classification problems. SVM is known for its high generalization capabilities, accurate classification, and simplicity in computation. For a given set of training samples, SVM builds an optimal separating surface with a linear or nonlinear mapping function, a normal vector, and the offset measure of the separating surface which is commonly known as hyperplane. A margin in SVM is the minimum distance between the support vector points and the hyperplane separating two classes, as depicted in Fig. 3. In addition to finding a hyperplane that separates the individual classes, a constraint is introduced that requires the hyperplane to exhibit the greatest possible distance between the closest training samples of each class (Ref. 28). This makes the optimization of SVMs a convex problem for which the global optimum can always be found (Ref. 29). Thus, as long as the relevant support vectors are contained within the training data, the optimization will always result in the same classifier, even if the training data are considerably reduced. The effect of reducing training data size can be visualized in Figs. 3(c) and 3(d), which shows the invariance of the SVM classifier with respect to the size of the training dataset. In comparison, Figs. 3(a) and 3(b) depict significant variance of linear ANN classifiers with respect to the size of the training dataset. It can be seen that SVMs are more robust to the size of the training data, and they generally exhibit better generalization performance than deep CNNs. Furthermore, in contrast to CNNs, SVMs can be computed much faster, enabling them to perform object detection on HD videos in real time even on embedded hardware (Refs. 30, 31).

However, SVMs exhibit one major drawback, which is their dependency on a strong feature extractor to provide the SVM with the relevant information required for solving specific classification or regression problems. Therefore, much of the prior research has focused on designing the most suitable feature extractors for various computer vision tasks.

Recent automatic object detection/classification techniques

Faster Region-Based Convolutional Neural Network (RCNN), YOLO (You Only Look Once), Single Shot MultiBox Detector (SSD), and hybrid methods are some of the prominent state-of-the-art computer vision techniques for automatic object detection/classification that we review here for completeness. Faster RCNN consists of three separate

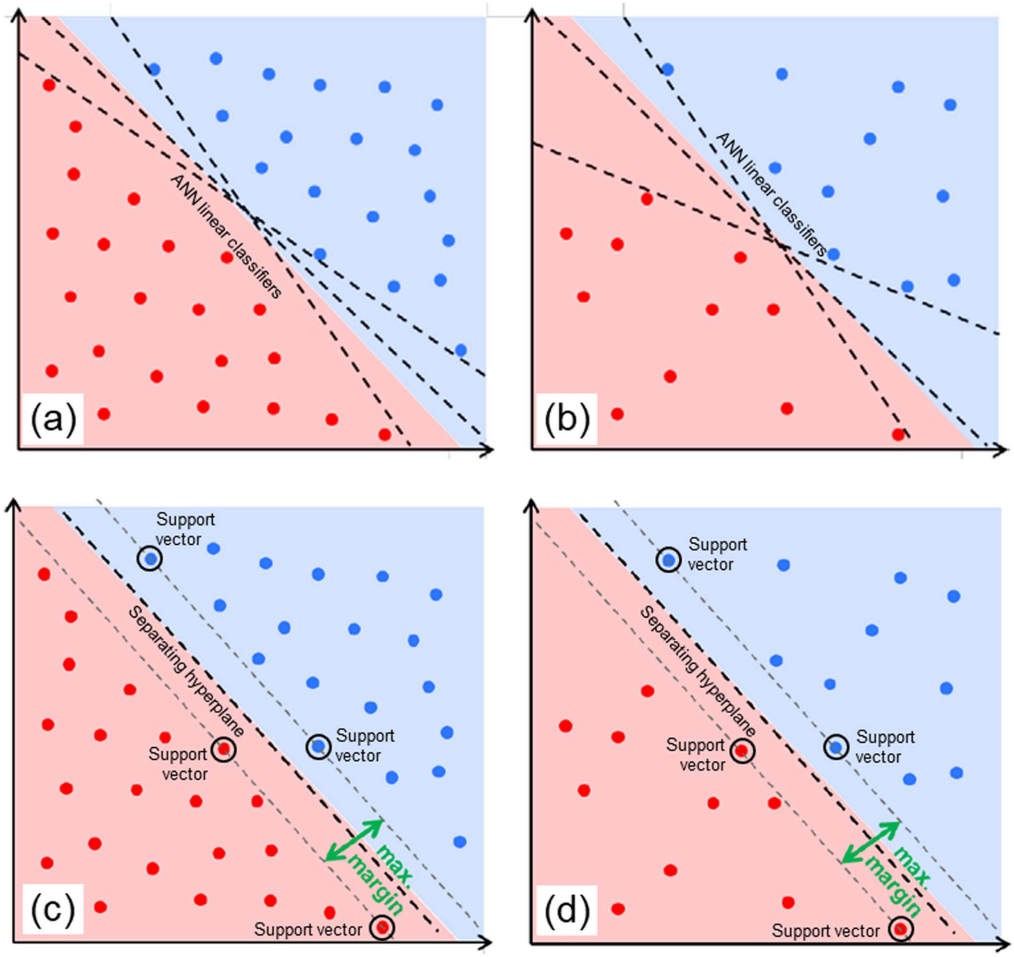

Fig. 3. Visualization of the effects of decreasing training samples on the generalization performance of ANNs (a) and (b) and SVMs (c) and (d) (Ref. 27).

networks to perform tasks like feature map generation (feature network), ROI prediction (region proposal network), and final detection and classification (final detection network) (Ref. 22). The input image is divided into 2000 ROIs that contain potential objects and a set of class predictions are made for each ROI. The advantage of Faster RCNN is that it gives accurate predictions when compared to its predecessors (RCNN and Fast RCNN in Ref. 32) and is robust to different sizes of objects (Ref. 22). The drawbacks of Faster RCNN are that it has a complex architecture with three different networks that are computationally expensive and hence the technique is very slow (Ref. 18).

YOLO is the current state-of-the-art for object detection problems (Ref. 18). The architecture of YOLO performs the tasks of feature extraction, informative region extraction, and classification in a single network which is similar to CNNs such as Visual Geometry Group (VGG) (Ref. 23), AlexNet (Ref. 17), etc. YOLO divides the feature image into S $\times$ S grid cells, and a set of anchor boxes of different sizes are generated for each cell. The class predictions are further made for the regressed bounding boxes. The advantage of YOLO is that it has a simpler architecture when compared to Faster RCNN and hence is faster and also more accurate (Ref. 18). The disadvantage is that the network is still complex as expensive operations like regression are performed to generate accurate results. There is also a faster version of YOLO with lesser layers but in this case the accuracy drops (Ref. 33).

The Single Shot MultiBox Detector (SSD) released in 2016 created new records for object detection tasks with outstanding performance and precision (Ref. 19). It is also a single network that performs the task of informative region extraction, detection of objects, and classifying them. Like anchor boxes in YOLO and ROIs in Faster RCNN, SSD proposes a set of fixed bounding box priors (approximately 1400) for every region that potentially contains object information. The bounding boxes are selected by non-maximum suppression and class predictions are made. The advantage is that SSD is the fastest of all the rest of the models, and its accuracy is on par with other methods. The disadvantage is that it is not robust to smaller objects and requires configuration of the prior boxes according to the desired object sizes, which adds complexity.

A hybrid object detection model based on the works of Refs. 20 and 27 combines a pretrained deep CNN (such as AlexNet in Ref. 17) and a powerful SVM classifier to perform the task of object detection. The input image is divided into a definite number of fixed-sized patches. The features are then extracted for each patch, and the SVM classifier performs the classification on each patch. Hence, a separate classifier is trained for each class (Ref. 20). The advantage of this hybrid approach is that it is less complex than the previously discussed techniques in that division of an image into fixed-size patches that contain potential objects reduces the complexity of the model. Replacing the fully connected layer of a CNN by an SVM improves the generalization performance as demonstrated by these studies (Refs. 20, 27). The disadvantage of this approach is that since the classifiers are fused together for multiclass classification, the complexity and the inference time are affected when the number of object classes to be identified is large.

Combining CNN and SVM for Automatic Obstacle Detection/Classification

In this section, we present key, high-level requirements for the automatic obstacle detection/classification problem and introduce our machine learning method to address the problem. We also present arguments in support of our choice of the method in comparison to the notable computer vision techniques discussed previously.

Key considerations for obstacle detection/classification

Although detailed functional requirements and installed performance will not be discussed, we summarize below some of the main considerations that guided our choice of learning-based obstacle detection and classification method:

1) Given the limited amount of open-access airborne imagery and the prohibitive costs of dedicated aerial data-gathering campaigns (especially without an industry-wide collaborative approach), the detection performance needs to be robust to the small sizes of training datasets.

2) The system needs to be modular so that equivalent detection performance can be readily assured with different types and suppliers of imaging sensors.

3) The training process needs to be implementable on desktop workstations with acceptable training times.

4) The inference process needs to be implementable on embedded, commercial off-the-shelf GP-GPUs with acceptable runtime performance.

Proposed method

In view of the aforementioned considerations, we propose a hybrid object detection model, that is a combination of CNN and SVM to leverage the strengths of each technique while overcoming their individual limitations for automatic obstacle detection/classification. This approach, previously introduced in Ref. 27, involves using a part of a pretrained CNN to extract distinctive features in regard to the obstacles of interest. This includes a cascade of several convolutional layers (deep CNN) as an implicit feature extractor at several layers of abstraction. Research has shown that CNNs that have been highly optimized for discriminating thousands of object classes on huge datasets can be successfully reused as feature extractors for domains they have not even been trained for (Refs. 34, 35). Relying upon this strong generalization performance of the CNN feature extractors, we feed the output of pretrained CNN feature extractors to our proprietary frequency-domain SVM classifier implementation described in Ref. 12 for robust and accurate obstacle classification.

The first benefit afforded by the present approach is that the combined CNN+SVM model drastically reduces training data requirements since the feature extraction part relies on CNNs that have been trained a priori on large open-source datasets, while our proprietary SVM obstacle classification requires only a small amount of training data for good generalization, as discussed previously. The high generalization performance of CNN+SVM with fewer data reduces the need to secure large training datasets in comparison to the state-of-the-art methods discussed previously. Furthermore, the drawback of CNN+SVM that it divides each input image into fixed-sized patches is acceptable since the obstacles to be detected do not vary significantly in size. Even the smallest-sized objects can be detected by defining object subclasses based on size (Refs. 20, 27).

The second benefit is in regard to the computational performance. A CNN typically consumes the most processing time to propagate data through the fully connected layers at the end of the network for object classification purposes. In contrast, the convolutional layers for feature extraction can be evaluated rather quickly using the parallel processing capabilities of modern GP-GPUs. By replacing the fully connected layers of CNNs (i.e., those responsible for classification) with our SVM classifiers known for their fast performance (Refs. 30, 31), the overall processing time for obstacle classification is significantly improved in comparison to some of the state-of-the-art methods.

In summary, our CNN+SVM model offers a highly efficient, generalizable, and computationally optimized method for onboard obstacle detection/classification using commercially available airborne optical sensors.

Experimental Setup

The proposed CNN+SVM approach has been previously evaluated on DARPA's Neovision2 dataset (Ref. 27). In the following sections, we assess its performance and demonstrate its capability for the helicopter obstacle detection/classification use case.

We performed the experiments in two steps. The first step involved model training and performance evaluations on a standalone desktop workstation hosting an NVIDIA Quadro K2200 (a mid-range GPU) and using different sets of prerecorded airborne video sequences in the visual and infrared spectral bands. This step included a training phase and an inference phase to assess detection and runtime performance. The second step involved system implementation and model testing/inferencing onboard an experimental H145 helicopter using a low-cost, low-weight, and small form factor NVIDIA Jetson AGX Xavier developer kit. For this second step, we reused the offline trained classifiers for online inference onboard the helicopter.

As the timescale for experimental flight testing on full-scale helicopters is typically long, the testing time limited, and the datasets proprietary, we conducted the experiments only with the proposed CNN+SVM method. Performance comparisons with other state-of-the-art methods described previously are beyond the scope of this work.

Sensor setup

The airborne sensor suite that delivered the data included two electro-optical sensor systems. The first system is a HeliEVS sensor developed by Elbit Systems, operating in the visible, near-infrared, and long-wave infrared spectral bands and providing automatic image fusion from the three spectral bands. The full datasheet of the camera is available in Ref. 36. Our choice of this camera was motivated by two reasons. First, Airbus Helicopters and Elbit Systems had entered into a cooperation agreement for joint research in DVE mitigation for helicopter missions. The research objectives, installation details, and main results of the DVE mitigation program have been summarized in Ref. 6. In the frame of this cooperation, Elbit Systems had supplied Airbus Helicopters with a HeliEVS prototype for experimental flight testing. That cooperation provided a ready-to-use camera installation on the experimental aircraft. Second, Elbit has designed the HeliEVS camera according to the applicable aviation industry design standards for vision systems (RTCA DO-315), environmental conditions (RTCA DO-160), and airborne electronic hardware (RTCA DO-254). These considerations are essential to achieving the applicable safety objectives and showing compliance with civil airworthiness requirements.

The second sensor system is a high-definition, commercially available visible spectrum flight-test instrumentation (FTI) camera called Atom-One manufactured by Dream Chip Technologies. The full datasheet of the camera is available in Ref. 37. The choice of this FTI camera was motivated by the fact that it is a standard installation during experimental testing at Airbus Helicopters. Being commercially available in the

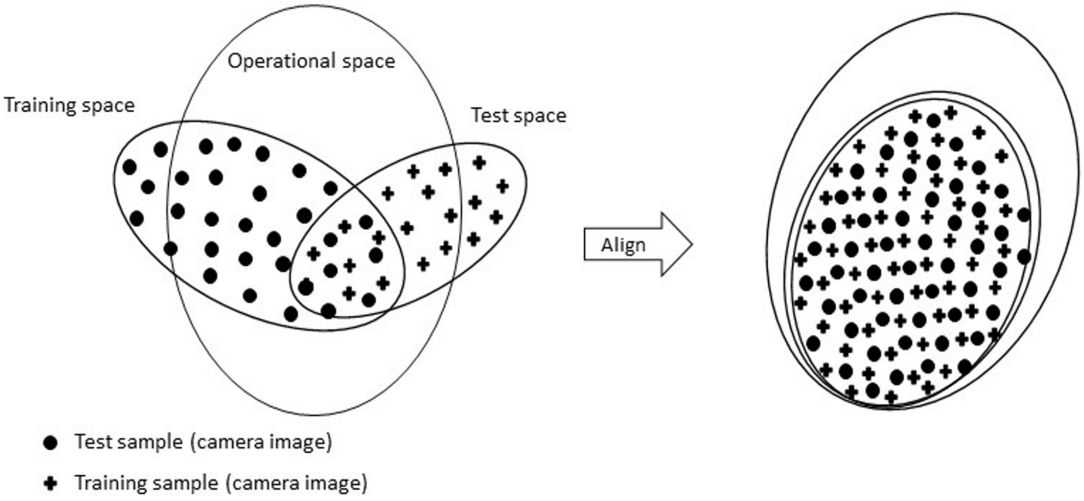

Fig. 4. Operational, training, and test spaces (Ref. 38).

consumer electronics market, it provides full high definition (1920 × 1080 pixels) at a low cost. Although the camera itself is not designed according to the aviation industry design standards, it has been qualified by Airbus Helicopters in regard to essential environmental requirements for installation on experimental helicopters. This latter aspect makes it a viable low-cost alternative in future airworthiness compliance demonstration activities requiring lower safety objectives.

Accordingly, the dual sensor setup allowed us to test the CNN+SVM model on cameras meeting different levels of performance and safety objectives for airborne applications.

Dataset generation

In regard to the datasets for machine learning, it is necessary to first outline the concepts of operational space, test space, and training space (Ref. 38). Operational space refers to the range of conditions in which the model operates and is expected to function as intended. Training space refers to the range of conditions for model training, and test refers to the range of conditions for model testing. For the automatic obstacle detection/classification to perform accurately, it is essential that all three spaces are aligned as discussed in Ref. 38 and shown in Fig. 4.

Due to the lack of publicly available datasets for objects that represent obstacles for low-altitude helicopter flight, we used airborne video sequences from proprietary flight test for gathering training data. These training data were generated as part of a flight-test program in a recently concluded DVE technology demonstrator project on the H145 helicopter between November 2017 and May 2018 in Ref. 6. These flight tests were conducted primarily in the Donauwörth area of Germany.

We manually extracted images depicting different obstacle types from the aforementioned airborne video sequences in different lighting conditions and different orientation angles. During the training phase, we manually annotated the image coordinates representing the obstacles of interest. The obstacle classes consisted of ground-fixed objects, namely wind turbines, antennas, pylons, chimneys, and TV towers. We defined an additional class for aerial obstacles that principally included helicopters.

We used a sufficiently similar training dataset for training the CNN+SVM model and test dataset for performance assessment to ensure alignment between the operational, training, and test spaces, to avoid outliers, and to draw meaningful conclusions from our experiments. We also assessed the generalization performance on a few selected visual and infrared video clips of the same set but originating from different real helicopter flights.

We note that the total number of training images per obstacle class was in the order of hundreds of images obtained from about a dozen hours of flight data recordings, which is significantly lower compared to the training dataset requirements of most state-of-the-art deep learning techniques for computer vision.

Desktop experimental setup

The first step of the experiment involved desktop setup for the training and inference of the CNN+SVM model, for which we used the Airbus Defense & Space proprietary software package for automatic target recognition described in Ref. 27 and installed on a standalone desktop workstation hosting an NVIDIA Quadro K2200 GPU. In this package, the implementation of the CNN feature extractor is built upon the Caffe Framework (Ref. 39), which has been developed by the Berkeley Vision and Learning Center and which provides a number of pretrained CNNs such as Alexnet (Ref. 17), GoogleNet (Ref. 24) or VGG (Ref. 23) that are highly optimized on large image datasets. The implementation of the SVM classifier is based on Ref. 12.

During the training phase, we divided each obstacle class into three subclasses for size (small, medium, and large) based on the average pixel size of the main class. This way, the SVM classification models can cope with some amount of scale invariance of the objects to be identified from different viewing ranges. For each obstacle class, we trained one SVM classifier according to the principle of one-versus-all (Refs. 20, 40), which is a scheme of using binary classification algorithms for multiclass classification. Finally, for a comparative assessment of the runtime performance of the CNN+SVM model, we also compared the results with published data on state-of-the-art object detection methods, namely Faster RCNN (Ref. 22) and YOLOv3 (Ref. 18), whose results we present in the next section.

Helicopter experimental setup

The second step of the experiment involved the onboard implementation and flight testing in the period May 2021 to August 2021, shown later in Table 5. Figure 5 shows the four components involved in the functional chain and illustrates the simplicity of the onboard implementation. Note that this simple implementation only considers a display device; no pilot command and control and alerting concept were considered at this stage.

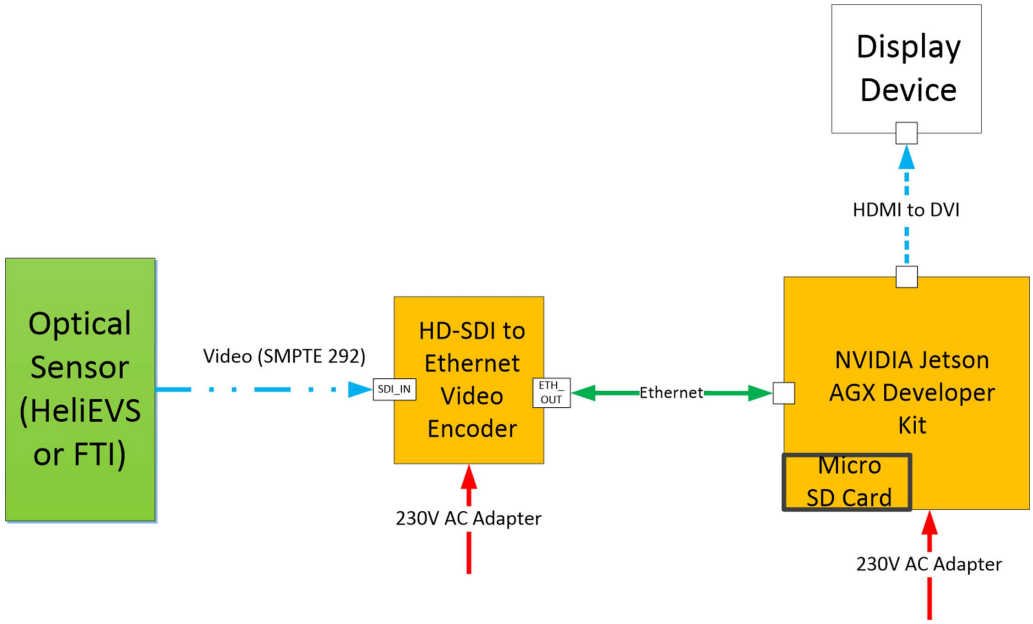

Fig. 5. Architecture for implementation in flight.

The optical sensor in Fig. 5 is either the HeliEVS camera or the FTI camera used interchangeably. Both sensors were installed externally on the nose section of the experimental H145 helicopter. The sensors transmit video frames in SMPTE 292 standard, which is encoded to H.265/H.265 format and transmitted via HTTP or RTP protocol over Ethernet to the NVIDIA Jetson AGX Xavier GP-GPU developer board shown in Fig. 6. The NVIDIA board hosts the real-time inference software for the CNN+SVM algorithm, which retrieves camera video frames via the Ethernet port, detects and classifies obstacles in the incoming image frames, superimposes bounding boxes and obstacle types, and transmits the resulting video sequences to the display device over HDMI.

Figure 7 shows the sensor installations on the helicopter. Figure 8 shows the experimental hardware setup installed on the aircraft cabin floor. Figure 9 shows the classified obstacles presented on a display device inside the cockpit cabin.

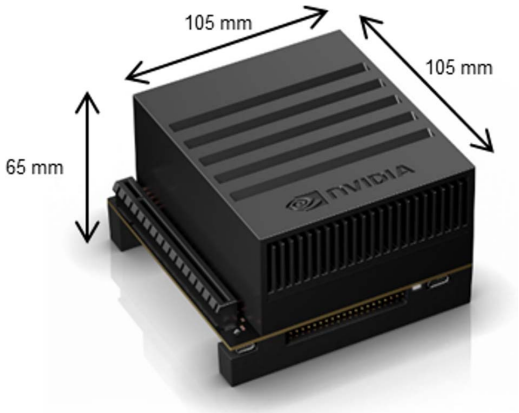

Fig. 6. NVIDIA Jetson AGX Xavier GP-GPU developer board.

Results and Discussion

In this section, we present quantitative and qualitative results of the CNN+SVM model from both desktop testing and flight testing. Due to the importance of the real-time capability for obstacle detection/classification, we present runtime performance figures achieved on a desktop workstation and the NVIDIA Jetson AGX Xavier GP-GPU installation on the helicopter described previously.

Figures 10–15 show the qualitative results of the CNN+SVM method using images in both visual and infrared spectral bands. These images are snapshots taken out of the video recording of the CNN+SVM inference algorithm running on the desktop workstation. The detection performance was satisfactory in all lighting conditions for which training data were available, in all viewing angles and spectral bands.

Detection accuracy

One quantitative criterion is the accuracy of the detection of each obstacle class in the visual and infrared spectral bands. In computer vision applications, accuracy is often measured in terms of the average precision and average recall for all instances of each obstacle class in the test dataset, along with the mean values of the average precision and average recall across all classes.

Precision is defined as the number of true positives (TP) divided by the sum of true positives and false positives (FP):

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall is defined as the number of true positives divided by the sum of true positives and false negatives (FN):

$$\text{Recall} = \frac{TP}{TP + FN}$$

Precision indicates what proportion of the objects detected by the algorithm is relevant, whereas recall indicates what proportion of all relevant objects the algorithm has actually detected. Precision and recall usually share an inverse relationship, and both are dependent on the model score threshold. Figure 16 illustrates how precision and recall are computed for a single camera frame and a generic test sample.

Table 1 presents the accuracy of the CNN+SVM model from desktop simulations. This result is based on a selective set of 50 test images for

Fig. 7. FTI and HeliEVS sensor installations on an experimental H145 helicopter.

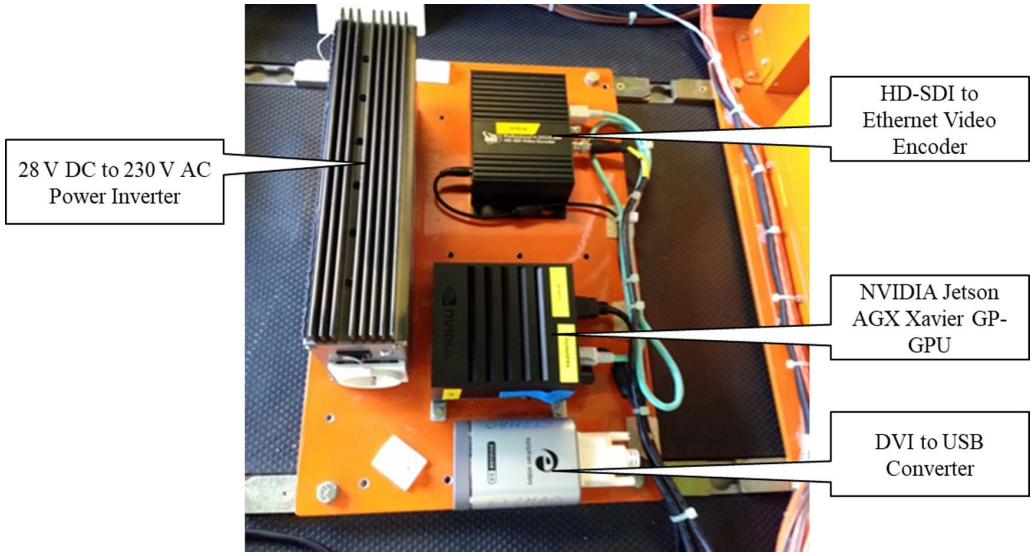

Fig. 8. Experimental electronic hardware installation on the cabin floor.

Fig. 9. Real-time display of the CNN+SVM output in the operator cabin console.

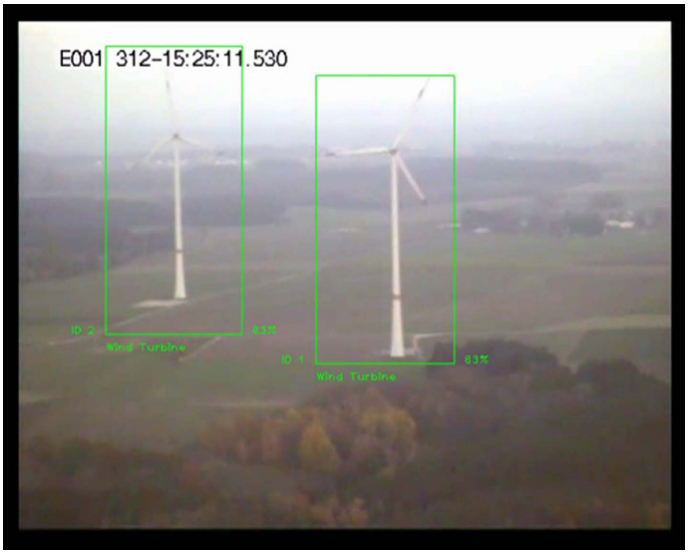

Fig. 10. Wind turbine detection from the FTI camera.

Fig. 11. Wind turbine detection from the HeliEVS camera.

Fig. 12. Chimney detection from the FTI camera.

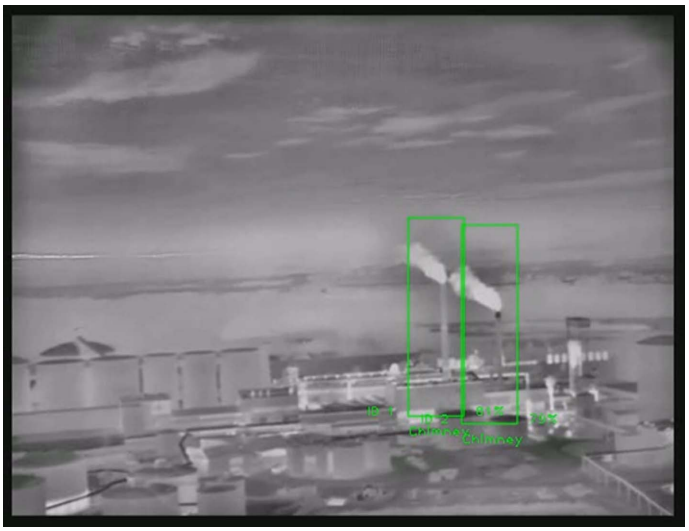

Fig. 13. Chimney detection from the HeliEVS camera.

Fig. 14. Antenna and helicopter detection from the HeliEVS camera.

each class so as to provide a common basis for all classes. To produce this result, we manually selected test images that cover all possible obstacle variations such as scale, size, perspective, and lighting conditions in the available dataset.

As seen in Table 1, the average precision and average recall for all classes are better than 90%. The mean average precision across all classes is 96%, and the mean average recall across all classes is 95%. From these results, we infer that the accuracy of the CNN+SVM method is satisfactory, considering that the model training dataset had only a few hundreds of images. However, we note that the accuracy can drop if the dataset has a larger number of images with greater variations in their appearance.

Based on the precision and recall values in Table 1, we can also infer that the accuracy is slightly better on the HeliEVS imagery than on the FTI visual imagery (cf. obstacle classes wind turbine and chimney in Table 1). The main reason for this observation is the effect of weather conditions on the visual spectrum images that reduces the clarity of the appearance of obstacles. By virtue of combining multispectral images, the effect of weather and daylight on HeliEVS is smaller. The HeliEVS

Fig. 15. TV tower detection from the HeliEVS camera.

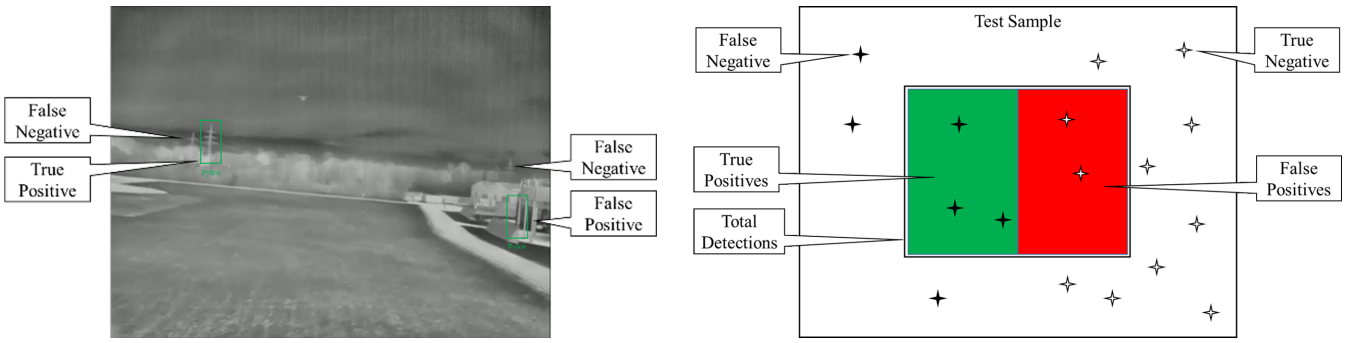

Fig. 16. Illustration of the elements used in the evaluation of precision and recall metrics: a camera image showing detection of pylons (left) and a generic test sample (right).

Table 1. CNN+SVM obstacle detection accuracy for desktop simulations

Classes	Camera Source	Ground Truth	True Positives	False Positives	Missed Detections	Average Precision	Average Recall
Wind turbine	FTI	50	48	3	2	0.94	0.96
Wind turbine	HeliEVS	50	50	0	0	1.0	1.0
Chimney	FTI	50	46	2	4	0.95	0.92
Chimney	HeliEVS	50	49	1	1	0.98	0.98
TV tower	HeliEVS	50	49	0	1	1.0	0.98
Antenna	HeliEVS	50	44	3	6	0.93	0.88
Helicopter	HeliEVS	50	47	2	3	0.96	0.94
Pylon	HeliEVS	50	48	5	2	0.91	0.96

sensor also provides a better contrast of the obstacles of interest against background clutter. The CNN+SVM algorithm was also able to detect partially occluded, overlapping, and smaller sized obstacles.

Training time and runtime performance

Another quantitative criterion is the performance of the obstacle detection in terms of the training time and the inference time. This measure is important considering that our requirements include the ability to train the algorithm with smaller datasets and to host the algorithm on airborne embedded devices onboard the helicopter.

We measure training time by the amount of time taken to train the complete set of images for each obstacle class. We measure inference performance by the number of frames of video processed per second on a target computing platform.

Table 2 presents the training times required to build each of the classifiers. We performed the measurement on the selected set of images per class as given in the table. The overall training time for all classes comprising 877 images is about 35 min. Given the small number of images and low training time, the CNN+SVM model is well-suited to carry out repeated experiments until the desired accuracy is achieved in wider operational domains. We estimate that testing the model in different operational environments would require at least a few thousands of images per class with all possible variations in the data based on the obstacle detection/classification mission requirements for good generalization.

Table 3 presents the inference performance of the CNN+SVM model on an NVIDIA Quadro K2200 GPU. It shows that the inference performance is high, which makes it suitable to run the model on flight-test videos in a real-time sense to obtain a list of detected obstacles in each frame.

Table 4 presents the runtime performance of the CNN+SVM model and compares it with two state-of-the-art methods running on an NVIDIA Jetson AGX Xavier board based on the results presented in Ref. 33. It can also be seen that the embedded performance of the proposed CNN+SVM approach is significantly better than the Faster RCNN and comparable to YOLOv3. This is because, unlike YOLOv3, our approach has a separate SVM classifier for each object class, and the number of classes affects the runtime performance of our system. However, we note that the training needs of the CNN+SVM model are far lower than YOLOv3.

Table 2. Training time to build the CNN+SVM model

Class	Camera Source	Number of Images	Training Time (min)
Wind turbine	FTI	112	3.56
Wind turbine	HeliEVS	124	4.02
Chimney	FTI	98	4.12
Chimney	HeliEVS	101	4.02
TV tower	HeliEVS	136	6.44
Antenna	HeliEVS	84	3.03
Helicopter	HeliEVS	102	3.22
Pylon	HeliEVS	120	5.51

Flight data analysis

Following a successful desktop experimentation, we conducted flight testing from May 2021 to August 2021 in an experimental H145 helicopter using the test setup described in the previous section. Here we provide qualitative and quantitative analysis of the onboard obstacle detection performance on an embedded NVIDIA Jetson AGX Xavier GP-GPU from live forward looking camera feeds. For these flight tests, we only connected the HeliEVS camera to the CNN+SVM model due to its better detection performance noted during desktop testing. We also recorded the video output of the NVIDIA Jetson AGX Xavier to be used later for postflight analysis. The classifiers we considered for these experimental flights correspond to obstacles that appeared most frequently during flight testing: wind turbine, pylon, and antenna. Moreover, we

Table 3. Inference performance of the CNN+SVM model on a desktop workstation hosting an NVIDIA Quadro K2200 GP-GPU

Number of Images	Model Performance (frames/second)	Total Inference Time (s)	Inference Time per Image (ms)
400	35	11.43	28.575

Table 4. Runtime performance comparison of the CNN+SVM model on an NVIDIA Jetson AGX Xavier GP-GPU board

Model	Runtime performance (frames/second)
Faster RCNN	1.3
CNN+SVM	10–13
YOLO V3	16–18

performed all flight tests under visual flight rules during daytime. Table 5 presents the environmental conditions prevalent during each of the three flight-test scenarios.

Onboard detection accuracy

We measured the onboard obstacle detection accuracy using the same precision and recall metrics described previously. The main point of interest of the flight testing was to assess the onboard detection accuracy for obstacles that are observed by the camera during different times of the day and from different camera viewing perspectives that may not be fully captured by the test datasets. Table 6 presents the detection accuracy of three classes: wind turbine, pylon, and antenna for the three aforementioned flight-test scenarios. Expectedly, we found that the average recall and precision values were lower than those observed during offline desktop simulations (comparing with Table 1). Yet, average precision greater than 0.85 indicates that false positives accounted for less than 15% of the overall obstacle detections. Likewise, average recall greater than 0.80 indicates that less than 20% of the obstacles remained undetected. We note that the detection performance will tend to worsen if the operating environment includes newer obstacle geometries, viewing angles, lighting conditions, and such variations that were not part of the training process. On the contrary, the detection performance will tend to improve as new observations are continuously fed into the training process to revise the model parameters.

Obstacle detection range

The detection range is an important criterion to measure how far obstacles can be reliably detected towards ensuring sufficient lead time to pilots for situational awareness. During flight testing, we asked the test pilot to approach each obstacle or a cluster of obstacles head on to ensure they remain in the camera field of view from as far away as possible. Table 7 shows the furthest distances at which each obstacle class could be reliably and correctly detected. It is clear that wind turbines were detected from the furthest distance whereas antennas had the lowest

Table 6. CNN+SVM obstacle detection accuracy during three flight-test scenarios

Flight Scenario	Object Class	Camera Source	Average Precision	Average Recall
Flight 1	Wind turbine	HeliEVS	0.927	0.900
	Pylon	HeliEVS	0.842	0.800
	Antenna	HeliEVS	0.947	0.900
Flight 2	Wind turbine	HeliEVS	0.894	0.850
	Pylon	HeliEVS	0.850	0.850
	Antenna	HeliEVS	0.882	0.750
Flight 3	Wind turbine	HeliEVS	0.936	0.880
	Pylon	HeliEVS	0.885	0.850
	Antenna	HeliEVS	0.940	0.800

Table 7. CNN+SVM maximum obstacle detection range from flight tests

Obstacle Class	Maximum Detection Range (m)
Antenna	685
Pylon	1280
Wind turbine	3386

maximum detection range. We also note that the maximum wind turbine detection range occurred when approaching from the front view so that three wind turbine blades were readily apparent. Since wind turbines have a unique shape and geometry (mast with three blades) that cannot easily be confused with other objects in the environment, its detection range was the highest. In contrast, an antenna having a single mast and devoid of peculiar features could only be correctly detected at about 685 m. These observations highlight the challenges faced by deep learning-based computer vision methods, namely the need for unique and distinguishable features and the availability of representative training datasets.

Future Work

Qualifying machine learning–based computer vision techniques for enhanced obstacle situational awareness on helicopter platforms has many open challenges—robustness, generalization capability, operational space expansion, trustworthiness, design assurance, data quality and quantity, among others—that should be addressed as part of future work. Recent research has led to concepts such as few shot obstacle detection (Ref. 41), which are a collection of learning techniques that, when trained on large, publicly available annotated datasets, are capable of detecting novel objects having smaller annotated datasets. Likewise,

Table 5. Summary of flight-testing scenarios

Flight	Date	Time	Outside Temperature	Weather	Visibility
Flight 1	June 17, 2021	0940–1020	24°C	CAVOK[a]	>3000 m
Flight 2	July 22, 2021	1345–1500	25°C	Partly cloudy	>3000 m
Flight 3	August 13, 2021	0800–0845	22.5°C	CAVOK[a]	>3000 m

[a]Ceiling And Visibility [are] OK

computer-generated synthetic data have shown great promise to augment the limited real-world training data. Recent works have investigated minimal sets of real-world training data for equivalent detection performance (Ref. 42) and new synthetic datasets for airborne applications (Ref. 43), advances that will help address the aforementioned challenges.

Conclusions

We have presented a hybrid automatic obstacle detection and classification method that is tailored to low-altitude helicopter missions. Our method leverages the potential of state-of-the-art deep CNNs for efficient obstacle feature extraction and pairs them with a proprietary high-speed SVM for obstacle classification. The proposed CNN+SVM obstacle detection and classification method has the following highlights:

1) It requires comparatively small amounts of training data which is particularly advantageous for airborne use cases.

2) It has a high generalization ability and detection accuracy.

3) It has light resource consumption in terms of training time and runtime performance on commercial off-the-shelf NVIDIA GP-GPUs.

4) It has a simple architecture (CNN+SVM) with fast convergence.

5) It gives good performance when the operational context is well-defined and the training datasets are representative of the testing conditions.

We have reported experimental results from both offline desktop-based testing and low-altitude flight testing with forward-looking imaging sensors and an airborne GP-GPU setup. Our offline training required about 35 min for 877 images covering eight different obstacle classes, which is significantly lower compared to contemporary state-of-the-art object detection methods. Moreover, our inference performance on the NVIDIA Quadro K2200 GPUs was 35 frames/second, while on a miniature NVIDIA Jetson AGX Xavier board it was 10–13 frames/second. We could achieve better than 90% detection accuracy from offline desktop-based testing measured using the average precision and average recall metrics, and better than 80% detection accuracy in flight tests, albeit on operationally limited datasets and scenarios. Furthermore, we have noted slightly better detection accuracy on the HeliEVS multispectral sensor, which provides an automatic fusion of visual, near-infrared, and long-wave infrared spectral bands, as compared to a purely visual FTI optical sensor.

The proposed deep learning approach holds promise for improved pilot situational awareness against potential obstacles in low-altitude missions and towards fulfilling Airbus Helicopters' aviation safety ambitions.

References

[1]Bernandersson, M., Eagles, T., Greiller, M., Masson, M., and Steel, J., "EHEST Analysis of 2006-2010 European Helicopter Accidents," EASA EHEST Report, August 2015.

[2]Vreeken, J., "Helicopter Flight in a Degraded Visual Environment," NLR-TP-2013-559, December 2013.

[3]SC-212, "Minimum Operational Performance Standards (MOPS) for Helicopter Terrain Awareness and Warning System (HTAWS) Airborne Equipment," RTCA DO-309, March 2008.

[4]Chandrasekaran, R., Payan, A. P., Collins, K. B., and Mavris, D. N., "Helicopter Wire Strike Protection and Prevention Devices: Review, Challenges, and Recommendations," *Aerospace Science and Technology,* **98**, 105665 (2020), DOI: 10.1016/j.ast.2019.105665.

[5]Waanders, T., Scheiblhofer, R., Qian, Q., van Noort, B., Ziegler, V., Schubert, F., and Koerber, R., "Helicopter Rotor Strike Alerting System," Proceedings of the 41st European Rotorcraft Forum, Munich, Germany, September 1–4, 2015.

[6]Waanders, T., Roth, F., Singer, B., Fadljevic, D., Plorin, J., Shpak, M., Hasharoni, O., and Limonad, A., "Integration and Flight Testing of a DVE System on the H145," Proceedings of the 45th European Rotorcraft Forum, Warsaw, Poland, September 17–20, 2019.

[7]Zoppitelli, P., Mavromatis, S., Sequeira, J., Anoufa, G., Belanger, N., and Filias, F., "Embedding Intelligent Image Processing Algorithms: The New Safety Enhancer for Helicopters Missions," Proceedings of the 44th European Rotorcraft Forum, Delft, The Netherlands, September 18–21, 2018.

[8]Böhm, H.-D. V., Frank, J., and Ehrenwinkler, R., "Flight Trials with the Mission Flight Aids of NH90 TTH," Proceedings of the 58th Annual Forum of the American Helicopter Society, Montreal, Canada, June 11–13, 2002.

[9]Davis T., and Centolanze, L., "Laser Obstacle Detection System Flight Testing," USAAMCOM TR03-D-49, September 2003.

[10]Zimmermann, M., Gestwa, M., König, C., Wolfram, J., Klasen S., and Lederle, A., "First Results of LiDAR-aided Helicopter Approaches During NATO DVE-Mitigation Trials," *CEAS Aeronautical Journal,* Vol. 10, 2019, pp. 859–874, DOI: 10.1007/s13272-018-0354-8.

[11]Takahashi, M. D., Goerzen, C. L., Whalley, M. S., Mansur, M. H., Schulein, G. J., Minor, J. S., Ott, C. R., and Morford, Z. G., "Full-Scale Flight-Test Results for a Rotorcraft Safe Landing Area Determination Algorithm for Autonomous and Piloted Landing Approaches," *Journal of the American Helicopter Society,* **63**, 042001 (2018), DOI: 10.4050/JAHS.63.042001.

[12]Schertler K., and Liebelt, J., "Automatic Learning Method for the Automatic Learning of Forms of Appearance of Objects in Images," U.S. Patent 9361543, 2016.

[13]Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., and Qu, R., "A Survey of Deep Learning-Based Object Detection," *IEEE Access,* Vol. 7, 2019, pp. 128837–128868, DOI: 10.1109/ACCESS.2019.2939201.

[14]Nine, J., Saleh, S., Khan, O., and Hardt, W., "Traffic Light Sign Recognition for Situation Awareness Using Monocular Camera," Proceedings of the International Symposium on Computer Science, Computer Engineering and Educational Technology, Dresden, Germany, July 17, 2019.

[15]Dollar, P., Wojek, C., Schiele B., and Perona, P. "Pedestrian Detection: An Evaluation of the State of the Art," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 34, (4), April 2012, pp. 647–655, DOI: 10.1109/TPAMI.2011.155.

[16]Long, X., "Understanding Object Detection in Deep Learning," November 19, 2018, available at https://blogs.sas.com/content/subconsciousmusings/2018/11/19/understanding-object-detection-in-deep-learning, accessed October 26, 2022.

[17]Krizhevsky, A., Sutskever I., and Hinton, G. E., "Imagenet Classification with Deep Convolutional Neural Networks," Proceedings of the 26th Annual Conference on Neural Information Processing Systems, Lake Tahoe, NV, December 3–6, 2012.

[18]Redmon, J., and Farhadi, A., "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018, DOI: 10.48550/arXiv.1804.02767.

[19]Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C., "SSD: Single Shot Multibox Detector," Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, October 8–16, 2016.

[20]Uçar, A., Demir, Y., and Güzeliş, C., "Object Recognition and Detection with Deep Learning for Autonomous Driving Applications," *Simulation,* Vol. 93, (9), September 2017, pp. 759–769, DOI: 10.1177/0037549717709932.

[21]Sengupta, S., Basak, S., Saikia, P., Paul, S., Tsalavoutis, V., Atiah, F., Ravi, V., and Peters, A., "A Review of Deep Learning with Special Emphasis on Architectures, Applications and Recent Trends," *Knowledge-Based Systems,* **194**, 105596 (2020), DOI: 10.1016/j.knosys.2020.105596.

[22]Ren, S., He, K., Girshick R., and Sun, J., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 39, (6), June 2017, pp. 1137–1149, DOI: 10.1109/TPAMI.2016.2577031.

[23]Simonyan K., and Zisserman, A., "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv preprint, arXiv:1409.1556, 2014, DOI: 10.48550/arXiv.1409.1556.

[24]Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., "Going Deeper with Convolutions," Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, June 7–12, 2015, DOI: 10.1109/CVPR.2015.7298594.

[25]Abramatic, J., and Faugeras, O., "Sequential Convolution Techniques for Image Filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. 30, (1), February 1982, pp. 1–10, DOI: 10.1109/TASSP.1982.1163840.

[26]Nielsen, M. A., *Neural Networks and Deep Learning*, Determination Press, 2015, Chapter 6.

[27]Kroll, C., von der Werth, M., Leuck, H., Stahl, C., and Schertler, K., "Combining High-Speed SVM Learning with CNN Feature Encoding for Real-Time Target Recognition in High-Definition Video for ISR Missions," Paper 1020208, Proceedings of the SPIE Conference on Automatic Target Recognition XXVII, Anaheim, CA, May 1, 2017, DOI: 10.1117/12.2262064.

[28]Mavroforakis, M. E., and Theodoridis, S., "A Geometric Approach to Support Vector Machine (SVM) Classification," *IEEE Transactions on Neural Networks,* Vol. 17, (3), May 2006, pp. 671–682, DOI: 10.1109/TNN.2006.873281.

[29]Scholkopf, B., Sung, K. K., Burges, C. J., Girosi, F., Niyogi, P., Poggio, T., and Vapnik, V., "Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers," *IEEE Transactions on Signal Processing,* Vol. 45, (11), November 1997, pp. 2758–2765, DOI: 10.1109/78.650102.

[30]Hahnle, M., Saxen, F., Hisung, M., Brunsmann, U., and Doll, K., "FPGA-Based Real-Time Pedestrian Detection on High-Resolution Images," Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, June 23–28, 2013, DOI: 10.1109/CVPRW.2013.95.

[31]Suleiman, A., and Sze, V., "An Energy-Efficient Hardware Implementation of HOG-Based Object Detection at 1080HD 60 fps with Multi-Scale Support," *Journal of Signal Processing Systems,* Vol. 84, September 2016, pp. 325–337, DOI: 10.1007/s11265-015-1080-7.

[32]Girshick, R., "Fast R-CNN," Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, December 7–13, 2015, DOI: 10.1109/ICCV.2015.169.

[33]Hossain, S., and Deok-jin, L., "Deep Learning-Based Real-Time Multiple-Object Detection and Tracking from Aerial Imagery via a Flying Robot with GPU-Based Embedded Devices," *Sensors,* **19**, 3371 (2019), DOI: 10.3390/s19153371.

[34]Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T., "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition," Proceedings of the 31st International Conference in Machine Learning, Beijing, China, June 22–24, 2014.

[35]Girshick, R., Donahue, J., Darrell, T., and Malik, J., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, June 23–28, 2014, DOI: 10.1109/CVPR.2014.81.

[36]Elbit Systems Ltd., "Clearvision HeliEVS," 2016, available at https://elbitsystems.com/landing/wp-content/uploads/2018/07/Heli-Clearvision.pdf, accessed October 26, 2022.

[37]Dream Chip Technologies GmbH, "Atom One Cameras Datasheet," 2019, available at https://www.dreamchip.de/fileadmin/user_upload/dct_flyer_ATOM_one_web.pdf, accessed October 26, 2022.

[38]Alsing, S. G., "The Evaluation of Competing Classifiers," Ph.D. Dissertation 4733, Air Force Institute of Technology, Dayton, OH, March 1, 2000.

[39]Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T., "Caffe: Convolutional Architecture for Fast Feature Embedding," Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, November 3–7, 2014, DOI: 10.1145/2647868.2654889.

[40]Rifkin, R., and Klautau, A., "In Defense of One-Vs-All Classification," *The Journal of Machine Learning Research,* Vol. 5, December 2004, pp. 101–141, DOI: 10.5555/1005332.1005336.

[41]Antonelli, S., Avola, D., Cinque, L., Crisostomi, D., Foresti, G. L., Galasso, F., Marini, M. R., Mecca, A., and Pannone, D., "Few–Shot Object Detection: A Survey," *ACM Computing Surveys,* Vol. 54, (11s), January 2022, pp. 1–37, DOI: https://doi.org/10.1145/3519022.

[42]Gastelum, Z., and Shead, T., "How Low Can You Go? Using Synthetic 3D Imagery to Drastically Reduce Real-World Training Data for Object Detection," SAND-2020-10244, September 2020, DOI: 10.2172/1670874.

[43]Shermeyer, J., Hossler, T., Etten, A. V., Hogan, D., Lewis, R., and Kim, D., "RarePlanes: Synthetic Data Takes Flight," Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision, Waikoloa, HI, January 3–8, 2021, DOI: 10.1109/WACV48630.2021.00025.