CEAS Aeronautical Journal (2025) 16:973–991
https://doi.org/10.1007/s13272-025-00883-6

ORIGINAL PAPER

Runtime monitoring of operational design domain to safeguard machine learning components

Christoph Torens[1] · Franz Jünger[1] · Sebastian Schirmer[1] · Pranav Nagarajan[1] · Simon Schopferer[1] · Dmytro Zhukov[1] · Johann Dauer[1]

Received: 19 November 2023 / Revised: 6 May 2025 / Accepted: 9 July 2025 / Published online: 11 September 2025

Abstract

To increase the autonomy of future air taxis, machine learning is necessary for a lot of areas such as vision-based tasks. However, the safety aspect of any machine learning application is of significant concern for users, experts, and certification authorities. To mitigate the risk to passengers or people on the ground, any machine learning-enabled component requires demonstration of rigorous compliance to safety and development assurance standards. Standardization organizations and authorities are currently developing and establishing new guidelines for the safe use of machine learning applications in the aviation domain. This work showcases the concept of runtime monitoring for enabling the safe integration of an example machine learning application in the urban air mobility context: the detection of humans during a landing approach of an air taxi via an onboard camera. Such an application may be useful in the context of autonomous landing to ensure that no person on the ground is endangered. In particular, the concept of operational design domain monitoring is discussed in the context of the recent European Union Aviation Safety Agency guidance and demonstrated in flight testing. The operational design domain monitor is composed of several sub-monitors that supervise different parameters in the operational domain and detect out-of-distribution inputs. Through the development of this component, this work further extends our work on safe operation monitor and runtime assurance for machine learning applications. The flight test results indicate that monitoring the operational design domain can support performance as well as the safety of the operation.

Keywords Urban air mobility · Machine learning · Runtime monitoring · Runtime assurance · Operational design domain

Abbreviations

AI	Artificial intelligence
ASTM	American Society for Testing and Materials
BEC	Battery eliminator circuit
CDO	City-ATM Demonstrator
CoDANN	Concepts of Design Assurance for Neural Networks
ConOps	Concept of operations
CSI	Camera serial interface
DDS	Data distribution service
DJI	Da-Jiang Innovations Science and Technology Co.
DLR	German Aerospace Center
EASA	European Union Aviation Safety Agency
FAA	Federal Aviation Administration
FPS	Frames per second
GCS	Ground control station
GPS	Global positioning system
HAG	Height above ground
HSV	Hue saturation value
IoU	Intersection over Union
LED	Light-emitting diode
LTE	Long-term evolution
mAP	Mean average precision
MAVLink	Micro Air Vehicle Link
MEMS	Micro-electro-mechanical systems
ML	Machine learning
NCS2	Neural Compute Stick 2
NED	North-East-Down
OD	Operational domain
ODD	Operational design domain
OOD	Out of distribution
RC	Remote control
ROS2	Robot Operating System 2
RTPS	Real-Time Publish Subscribe

✉ Christoph Torens
christoph.torens@dlr.de

[1] Institute of Flight Systems, German Aerospace Center, Braunschweig, Germany

SOM	Safe operation monitor
SSH	Secure shell
UA	Unmanned aircraft
UAM	Urban air mobility
UAS	Unmanned aircraft system
VPU	Vision processing unit
WGS-84	World Geodetic System 1984

1 Introduction

With recent advances in artificial intelligence (AI) and machine learning (ML) there is an increasing appeal for their use even for safety-critical applications in the aviation domain. But the safety of any ML application is still a huge concern for experts. In the DLR project, "HorizonUAM" [1, 2], the safe autonomy work package [3, 4] targets the research of safety aspects of ML in the context of Urban Air Mobility (UAM), i.e. transportation services via air taxis in urban environments. This paper further extends our research on that topic [5].

A key research question is how ML-enabled autonomy can be safely applied in the context of UAM [6]. This poses a particular challenge for ML algorithms, including deep neural networks since standards like DO-178 are not suitable in this context. In particular, they do not account for the dependency on data. New guidance is required to enable the safe integration of ML in the aviation domain. Such safety standards for ML applications are currently under development. The European Union Aviation Safety Agency (EASA) has developed a first guidance for the certification of ML [7, 8]. In this paper, we utilize some of the key concepts of these guidelines to safeguard an ML component, the focus will be on operational domain (OD) and operational design domain (ODD). Additionally, the concept of out of distribution (OOD) will be discussed. These concepts will be used and explained with a UAM use case and evaluated in flight tests using a demonstrator Unmanned Aircraft (UA). Additionally, the relevant objectives of the EASA guidance will be detailed and explained in the context of the use case.

1.1 Use case

In our HorizonUAM project, the use case involves an air taxi landing on a vertiport. An ML component uses an onboard camera to detect if there are any persons on the vertiport. This algorithm is a state-of-the-art object detector for detecting humans on images [46] captured by an UA's onboard camera. The landing can not be initiated if there are any persons that could be harmed on the vertiport. In that case the landing has to be aborted and for example an alternative vertiport has to be used.

This use case was demonstrated in in flight tests at the DLR National Experimental Test Center for Unmanned Aircraft Systems in Cochstedt. Fig. 1 which shows the demonstration of an aborted landing during our flight tests of the simulated air taxi using our demonstrator UA.

1.2 Paper structure

The remainder of this paper is structured as follows. After the introduction, related work is discussed in Sect. 2, specifically the concepts of OD and ODD that have been introduced by the new EASA documents. Sect. 3 is dedicated to an explicit definition of the ODD. This is followed by the presentation of the software setup, specifically the ML component and architecture of the monitoring architecture in Sect. 4. Then, Sect. 5 presents the setup and the description for the flight test. In Sect. 6, results of the flight test are shown, including an evaluation of the ML component and an evaluation of the ML safety aspects. Next, Sect. 7 discusses OD and ODD objectives from the EASA guidelines for certification in context with the use case. Finally, Sect. 8 summarizes the work and gives a future perspective.

2 Related work

2.1 ML safety assurance

Recently, a literature review on the topic of ML safety [9] and an analysis of the existing and new regulations and guidance on ML safety have been published [10] by the authors. This work is a follow-up and extension to that aforementioned research. This field has a lot of interest from

Fig. 1 Demonstration of an aborted landing at a vertiport, due to persons on the ground during our HorizonUAM project. The persons are simulated by puppets

researchers, standardization groups and also certification authorities. The overall framework and the main context for verification and safety of ML in aviation is set by the following documents. The EASA published a number of documents regarding the safe use of ML, starting with the AI roadmap [11, 56], and following up with more details on different aspects of ML safety with Concepts of Design Assurance for Neural Networks (CoDANN) [12], as well as CoDANN II [13, 55], and a resulting Concept Paper First Usable Guidance for Level 1 Machine Learning Applications [7]. Finally, in 2024, EASA updated the guidelines with the Concept Paper First Usable Guidance for Level 1 & 2 Machine Learning Applications [8]. Furthermore, recently a report was published by the Federal Aviation Administration (FAA), Neural Network Based Runway Landing Guidance for General Aviation Autoland [14]. In addition to authorities such as EASA and FAA, also standardization organizations are currently working on standardization of AI in aviation, such as the joint SAE/EUROCAE working group WG-114 [15]. The output of this working group will be the document SAE ARP 6983 / EUROCAE ED 324. Some preliminary concepts detailing partial compliance regarding ODD definition and data design is shown in [16].

Research, specifically regarding the certification of ML in relation to existing traditional standards, such as DO-178C, has also been done recently [17–19]. Here, a combination of architectural mitigations and a mapping of DO-178C objectives to ML specific customizations are used to achieve design assurance level C. More recently [20] argues, that for most low-risk applications of ML the existing compliance framework should be enough to establish certification approval.

2.2 Operational domain

The concept of the operational domain (OD) was introduced only in the latest version of the EASA guidelines [8]. It should be noted, that the definition of OD is given in the context of the system and the Concept of Operations (ConOps). Since we are using the concept of OD and ODD in this paper, we will directly quote the definition from the EASA document:

> Operational domain (OD) – Operating conditions under which a given AI-based system is specifically designed to function as intended, in line with the defined ConOps. For instance, in the airworthiness domain, the Certification Specification for large transport aircraft, CS 25.1309 requires the identification of 'the aeroplane operating and environmental conditions'. A definition of 'foreseeable conditions' can be found under AMC 25-11 and generalised: 'Foreseeable Conditions - The full environment that the [...] system is assumed to operate within, given its intended function. This includes operating in normal, non-normal, and emergency conditions.' ([8], sect. G.1, page 246)

The document further acknowledges the fact that capturing the operating conditions is already a practice in the aviation domain, i.e. ConOps. However, this process is not formal enough for the development of ML. An additional aspect for differentiation between OD and ODD is that the OD may be formulated on a higher abstraction level (aircraft level), while the ODD (definition, see below) might be refined to additional, derived parameters, or parameter boundaries might be more stringent. The idea of a formal description of aspects of the ConOps for unmanned aircraft has been discussed in [21–23].

2.3 Operational design domain

Moreover, there is research regarding the specific topic of the operational parameters, also called ODD for the safety during operation. The concept of ODD was first used in the automotive domain [24]. Additionally, taxonomies for operational design domains are discussed in various automotive standards [25, 26]. Therefore, there is a lot of related research on this concept and specifically in relation to safety [27–32] in the automotive domain. There is recently also a growing interest in research of the concept of ODD in the aviation domain. Since our earlier work, beginning of 2023 [5], several works have been published. The path from ODD to scenario generation and simulation was proposed [23]. Also a data centric approach to characterize the ODD is given by [33]. Their framework discusses the categories of data relevant to defining and testing ML systems, such as Nominal, Outlier, Edge Case, Corner Case, Inlier, and Novelty data.

This paper is an example of a direct application of the ODD concept in the aviation domain on a demonstrator UA. This concept will be utilized to check the inputs of the ML algorithm in Sect. 6.2 and further discussed in Sect. 7. The first usage of ODD in the aviation domain was that the definition of ODD is given in the EASA document [7]. However, the latest version of the EASA document gave an updated definition for the concept of ODD:

> Operational design domain (ODD) – Operating conditions under which a given AI/ML constituent is specifically designed to function as intended, including but not limited to environmental, geographical, and/or time-of-day restrictions. The ODD defines the set of operating parameters, together with the range and distribution within which the AI/ML constituent is designed to operate, and as such, will only operate nominally when the parameters described within the ODD are satisfied. The ODD also considers dependen-

> cies between operating parameters in order to refine the ranges between these parameters when appropriate; in other words, the range(s) for one or several operating parameters could depend on the value or range of another parameter. ([8], sect. G.1, page 246)

The definition from EASA utilizes the term constituent, however for clarity we will use the term ML component throughout the paper.

2.4 Out of distribution

A similar concept is that of OOD. The concept of OOD is a commonly researched metric for assessing inputs that have variance along a distribution. Analyzing the OOD can increase the reliability of AI algorithms [34, 35]. In general, the basic idea of OOD monitoring is to check if the input data during operation is consistent with the trained data. The training data is analyzed on specific scalar metrics of (multidimensional) inputs. A simple example of such a metric is the value of the current altitude of the UAS. For more complex inputs, such as images, one example metric would be the brightness of an image. For all training inputs of this parameter, the number of occurrences of any value of the corresponding metric is counted and thus a distribution is established. Then, during operation the new input is compared to this distribution. For the safety of the ML component it is important to supervise ODD and OOD aspects. In fact the EASA guidance document [8] requires with its objectives to supervise both concepts, as discussed in the following sections.

> Out of distribution (data) – Data which is sampled from a different distribution than the one of the training data set. Data collected at a different time, and possibly under different conditions or in a different environment, than the data collected to create the ML model are likely to be out of distribution.

Additionally, the EASA CoDANNII report [36, 55] details the aspects of out of distribution (OOD) detection.

2.5 Runtime assurance

Finally, there is research on safety monitoring and Runtime assurance (RTA), which is an underlying principle, also for verifying the ODD during the operation [37, 38]. From EASA, the ForMuLA report [39] discusses the runtime monitoring of ODD as a formal method. In combination, these concepts can be used to safeguard the ML component against unsafe inputs. The idea is to supervise if the input that is fed to the ML component is consistent with the intended operating conditions and limitations. RTA, as described in ASTM F3269-21 [37], can be used as an alternate certification strategy when a complex function cannot be assured at designtime, i.e. proven to be safe prior to operational deployment through compliance to standards for system, hardware and software development assurance such as ARP 4754A, DO-254 and DO-178C. In the architecture proposed by the ASTM standard practice, a safety monitor supervises the outputs of the complex function to ensure they do not lead the system to unsafe states. If the outputs exceed a certain threshold, defined a priori, the safety monitor switches control of the system to a suitable recovery function through an RTA switch. Nagarajan et al. [38] discuss the concepts that went into the development of the standard practice as well as some challenges and gaps yet to be addressed. Most applications of the standard deal with the control function aspect of the architecture [40–43], yet there is little discussion in literature on the use of RTA for securing the inputs to complex functions. Any discourse on input monitoring is usually limited to the concepts of dynamic consistency checking and data conditioning. Even where there is work on the use of RTA for AI-based systems using video [44] or image [45] inputs for monitoring the complex controller, there is no explicit discussion of the concept of ODD monitoring within an RTA architecture. In this paper, we present how the concepts of RTA as presented in the ASTM architecture can be expanded to include input monitoring, especially using ODD and OOD monitoring to ensure safe operation.

In [44], Cofer et al. demonstrate an automatic aircraft taxiing system called TaxiNet, which uses a learning-enabled runway centerline tracking controller bounded by an RTA architecture. Here, they use computer vision (CV) along with position and inertial reference sensors to monitor large deviations from the centerline, however they do not provide many details on the validity checking of the video inputs they process for the CV-based monitor. In a more recent application of the ASTM F3269 architecture, He and Schumann et al. [45] demonstrate the application of a deep neural network based autonomous centerline tracking system through the use of images from cameras mounted on the aircraft. Here, there is more discussion of how parameters such as time of day, image fuzziness, dirt etc. could affect the performance of the image-based deep neural network. They use a temporal logic-based runtime monitor called R2U2 to then detect the performance degradation and switch to either a different neural network or a traditional reversionary controller. However, they specify fixed bounds on the above-mentioned parameters in their formal specification of the R2U2 monitor and do not use the concepts of ODD and OOD explicitly. In our work, we intend to show how ODD monitoring and OOD monitoring can be used both offline during the training process to determine the limits of performance for the ML-based components as well as how

these concepts can be used in runtime for safe operation monitoring Fig. 2.

3 Definition of OD, ODD and OOD

In this section we define an OD/ODD that will be supervised by our monitor for our specific use case. Below we provide the boundary values for the ODD properties during the flight tests, along with observations of their effects on ML performance for human detection. The ODD was set up to these values based on the available training data and engineering judgment.

3.1 Definition of OD

- OD
- Altitude: 0m - 100m
- Velocity: 0 m/s - 15 m/s
- Camera angle: -45° - +45°
- Geofence: Cochstedt airport airfield
- Daylight conditions
- Good weather

The OD defines the operating conditions of the operation. For simplicity reasons the operating conditions are defined to daylight conditions in good weather at the airfield of the Cochstedt airport. The flights take place below the attitude of 100 m at a maximum speed of 15m/s. To differentiate between OD and ODD, it should be noted that the OD define the operating conditions of the aircraft, while the ODD define the operating conditions of the ML component.

3.2 Definition of ODD

- ODD
- Altitude: 20 m to 50 m
- Velocity: 0 m/s to 10 m/s
- Camera angle: − 10° to +10°
- Geofence: 4 partly overlapping geofences, depending on the mission phase. Cochstedt airport airfield.

Fig. 2 Visual explanation of the concept of the ODD

Altitude: Our design idea was to take image data for altitudes between 20 to 50 m. The altitude is important, since it impacts the ground sample distance (GSD), which in turn determines the resolution of the images captured. At higher altitudes, objects such as people appear smaller and are represented by fewer pixels, which can impact the ability to identify and analyze them.

Velocity: We operate our drones regularly with velocities between 0 m/s (hover) and 10 m/s. First tests resulted in clear images, therefore we went with this range for the ODD. The velocity of the aircraft during data collection is a key factor that influences image quality. High speeds can lead to motion blur, which may degrade the image resolution and accuracy of object detection. Conversely, lower speeds can yield clearer images but may reduce the area covered during the flight. It is essential to balance velocity with the need for precise imagery to ensure optimal data collection for analysis.

Camera angle: Our drone is equipped with a downward-facing camera, which is optimal for capturing overhead imagery. Despite this fixed position, the drone's movements, including acceleration and deceleration, can cause the camera angle to vary. Our first estimates were to expect a range of $-10°$ to $+10°$. The camera angle has a huge impact on the captured images for person detection, since it alters the angle and thus the pixels of persons visible to the camera on the image.

Geofence: The geofence is a mitigation for operational safety [57]. Therefore, there is a safety benefit directly from utilizing a geofence. However, it can also affect the image in the sense that for the purpose of these tests flights, the geofence was edited to contain only a small area on the airfield of the Cochstedt airport. Here, we could assume consistent, monotonic background. A known limited operating area can support the image processing and analysis tasks. As soon as the airfield is left and grass and trees as well as streets, cars and houses are entering the image, the training and image processing obviously become more complex.

3.3 Definition of OOD

- Image quality (Static limits on image parameters)
- Brightness
- Saturation
- Entropy
- Edges

- OOD (Training distribution of image parameters)
- Brightness
- Saturation
- Entropy
- Edges

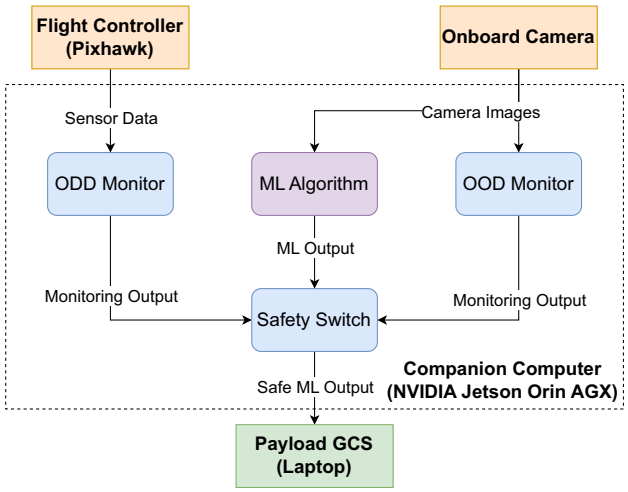


Fig. 3 Hardware (angular) and high-level software (rounded) components of flight test setup

Image quality: We considered four main image properties, such as brightness, separation, entropy, and edges. The idea was to simply cut off images on a fixed limit of these values to ensure uniform image quality. The images taken during test flights were within good daylight conditions. Brightness impacts the overall visibility within the image, while contrast affects the distinction between objects and their surroundings. Entropy measures the level of disorder or complexity in the image, and edges can similarly give a metric on image complexity.

Image training: Static boundary values for image parameters as discussed in the ODD aspect of "Image quality" can already help to filter out extreme cases, dark images, or extremely blurred images, see Fig. 13. However, the OOD is also important, this means to consider the specific ML component and the training it received. Images that are outside of the training distribution should be filtered out, similar to the other parameters of the ODD.

4 Software setup

To increase the safety of the ML-based person detection, a software architecture that incorporates additional monitoring has been developed. A high-level overview of the software architecture is depicted in Fig. 3. The main goal of this setup is to feed the images from the onboard camera into a ML algorithm which is able to detect humans on the ground. Then, the output from the ML algorithm is transmitted to the payload Ground Control Station (GCS) as a video stream. Additionally, two monitoring components as well as a safety switch have been added to increase the safety and trustworthiness of the ML algorithm. A more detailed explanation of the individual software components follows in the subsequent sections.

4.1 ML onboard person detection

The ML algorithm is a core component of the software architecture and is responsible for the onboard detection of humans in aerial images. As depicted in Fig. 3, the ML algorithm directly reads frames from the connected camera and uses object detection to determine bounding boxes for each person present in the current frame. The object detection is implemented via a neural network based on the YOLOv7-tiny architecture [46], a reduced version of the YOLOv7 architecture with lower performance requirements. This network is trained using aerial images from the HERIDAL database [47]. Additionally, recorded images from previous flight tests with a similar use case are added to the training data. The size of the input images has been customized to 1280 x 960 pixels. All of the training has been done using the darknet framework[1] for a better integration into existing inference source code, which enables support for older YOLO versions, such as YOLOv4 [48], and alternative inference hardware, e.g., Neural Compute Stick 2 (NCS2). The darknet weight and configuration files are directly supported by OpenCV,[2] allowing to load and execute the trained model with OpenCV directly. The neural network runs on the built-in GPU of the companion computer. During inference, the trained model achieves to process frames at a frame rate of up to 15 FPS. In many real-life scenarios where an onboard human detection could be deployed, e.g. to assess a potential landing area or a cargo drop zone, such a frame rate should be more than sufficient as the UA could hover above the target area for a few seconds before proceeding with the actual flight task. In that scenario, the onboard ML algorithm could assess the scene multiple times and should be able detect any relevant objects.

4.2 ODD monitor

This component monitors the operating conditions based on a specified ODD. This ODD monitor utilizes the sensor data from the payload Pixhawk to monitor operating conditions such as geofence, altitude, velocity and camera angle. As shown in Fig. 4, the ODD monitor consists of multiple low-level software components, which are subsequently described in more detail.

The ODD described in Sect. 3 states that altitude, velocity and camera angle are part of the ODD.

[1] https://github.com/AlexeyAB/darknet.

[2] https://opencv.org/

4.2.1 Monitoring system state

This subsection describes the monitors for altitude, velocity and camera angle. In this case, for each parameter, simple boundary values are specified. The data comes from data streams, see Sect. (4.2.2) that are fed into a module that wraps a RTLola interpreter into the ROS2 environment. RTLola is a framework for monitoring of stream-based data [49].

4.2.2 MicroRTPS messaging

All the software components are implemented as Robot Operating System 2 (ROS2) modules. This enables a modular structure as well as reliable communication between the software components. By default, the Micro Air Vehicle Link (MAVLink) messages generated by the payload Pixhawk can not be distributed as ROS2 messages. However, the PX4 software includes a Data Distribution Service (DDS) interface which is also referred to as the microRTPS bridge. This bridge implements the Real-Time Publish Subscribe (RTPS) protocol which provides publisher-subscriber communication similar to the messages in ROS2. The usage of the microRTPS bridge requires to run a software called microRTPS client on the payload Pixhawk and its counterpart, the microRTPS agent, on the companion computer. The microRTPS agent receives sensor data from the payload Pixhawk and provides it as ROS2 messages for the other software components. The message types that should be supported by the microRTPS bridge can be specified according to the use case. For the geofence and altitude monitoring we require the message types containing the global position in (NED) coordinates and the local position in (NED) coordinates. In contrast, the velocity monitor and the camera angle monitor require messages with the current velocity vector and the attitude of the aircraft, respectively.

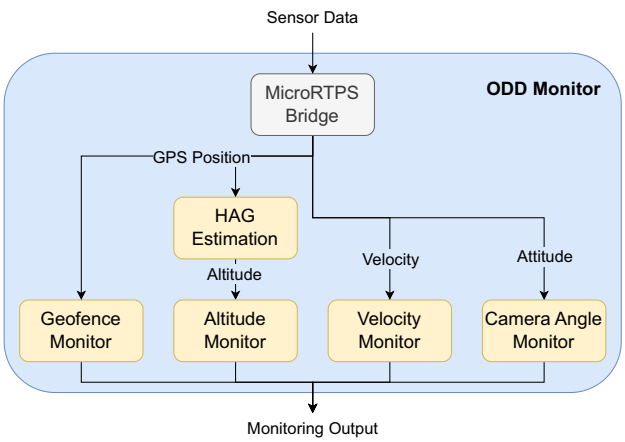


Fig. 4 Low-level software components of the ODD monitor

4.2.3 HAG estimation

The neural network is trained with images that have been taken at specific altitudes. To ensure the safe functionality of the object detection algorithm, the current altitude is compared with the altitudes of the images that have been used in the training process. For example, if images with altitudes between 20 and 50 m have been used in the training process, the object detection might not work properly below 20 m and above 50 m. As mentioned in Sect. 4.2.2 the payload Pixhawk outputs a global position in WGS-84 coordinates. However, the altitude in this global position is given in reference to the WGS-84 ellipsoid and does not represent the distance to the ground. To obtain the altitude above ground, the altitude monitor loads a terrain elevation map from the flight area. With this map, the altitude above ground can be computed by subtracting the terrain elevation at the current position from the WGS-84 altitude. The resulting height above ground is then provided for the altitude monitor as shown in Fig. 4.

4.2.4 Geofence monitor

Information about the operating environment are essential for composing an adequate dataset to train a neural network. For instance, training data showing humans on grass or fields would not be well suited to train for the detection of humans in an urban environment. Yet, if assumptions about the operating environment are made, they need to be checked *at runtime* to prevent entering a flight area where unforeseen features might occur. To reduce the likelihood of these unforeseen features and a shift in input distribution, the geofence monitor is used.

A geofence represents virtual barriers in space that the system under scrutiny is not allowed to cross. If the operation consists of different tasks that can be spatially separated it makes sense for an improved situational awareness to have overlapping geofences where a crossing represents the traversal from one task to another. Figure 5 depicts four overlapping geofences that were used for the flight test, see Sect. 5. *Geofence 1* contains the first vertiport, *Geofence 2* contains the second vertiport, *Geofence 3* contains the container cluster, and in *Geofence 4* the waypoint mission stopped. Each subplot represents the point of view of a respective geofence highlighted in blue. The flight path is depicted in green, orange, and red referring to inside geofence, close to geofence violation, and outside geofence, respectively. The figure shows that the waypoint mission was correctly tracked by the geofence monitor. The mission first started in *Geofence 1*, moved to *Geofence 2*, then to *Geofence 4*, then to *Geofence 3*, and finally ended up in the initial geofence.

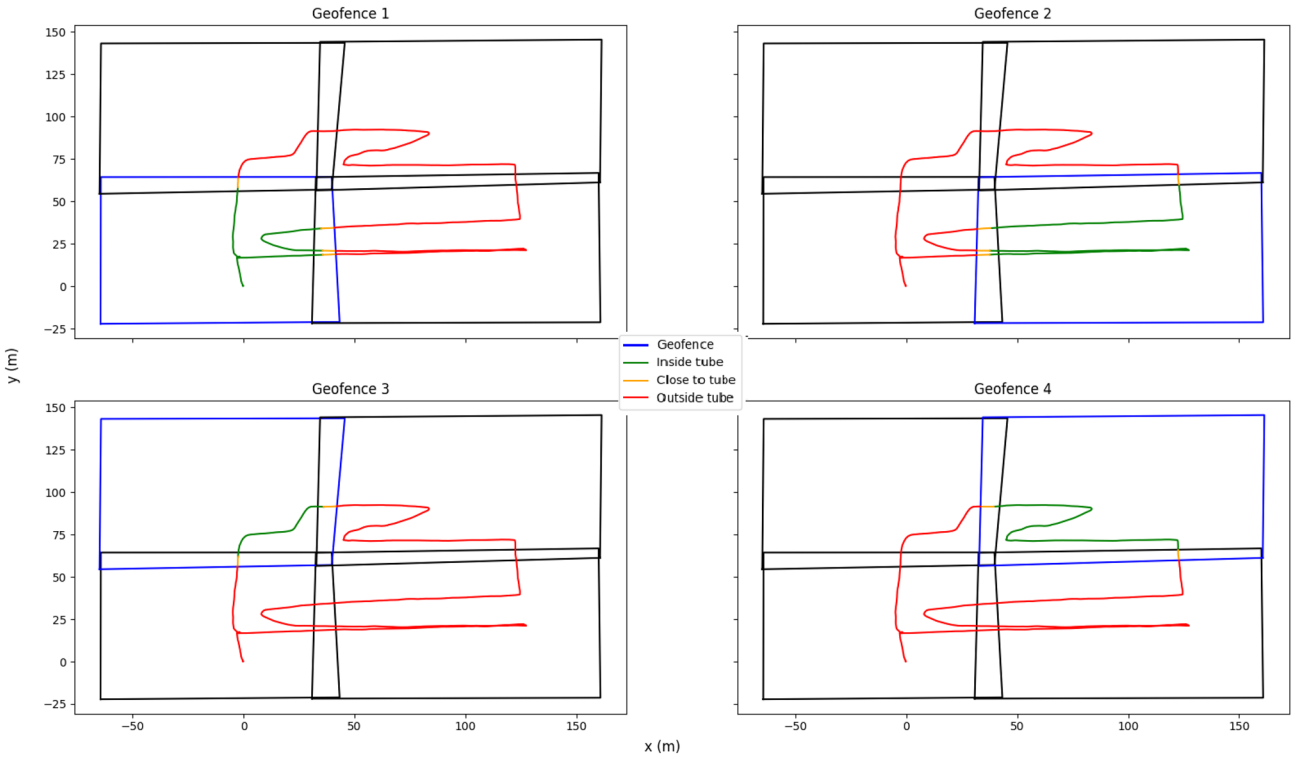


Fig. 5 Top-view of the overlapping geofences used for the flight test. In each subfigure, the monitor's outputs are given for the respective geofence highlighted in blue where green, orange, and red represent inside, close to crossing, and crossed monitor evaluations

Complementing the geofence of the operation area, geofencing of the planned waypoint mission was also used, which we refer to as tubes. Figure 6 shows the tubing results for our flight test. The pre-planned waypoint mission is depicted in blue. The flight is indicated by a colour varying line where green, orange, and red indicate when the UA was inside, near to the border of the tube, and outside of the tube, respectively. The numbers in the figure that range from zero to nine reflect the order of the waypoints. After the last waypoint is reached the waypoint mission stops. The figure shows that we were able to follow the pre-planned waypoint mission, i.e. , we stayed inside the tube. Only when the last waypoint was reached and the UA returned to the home position we did leave the tube. Besides tracking the position within the tube, we also notified when too much time expires until the next waypoint was reached in respect to a ϵ distance to the position of the waypoint. This ensures progress towards reaching the target waypoint, where ϵ provides a margin of tolerance around it, i.e., $|\text{UAS}.position - \text{waypoint}.position| < \epsilon$. When the UAS reaches the current target waypoint, the timer resets to track the next waypoint thereafter. If the timer expires, the current target waypoint was not reached in time. This can be seen in Fig. 7 where brighter colours indicate a larger value of the timer. For instance, the timer was reset at *Waypoint 3* but it took too long to reach *Waypoint 4*. Further, the timer

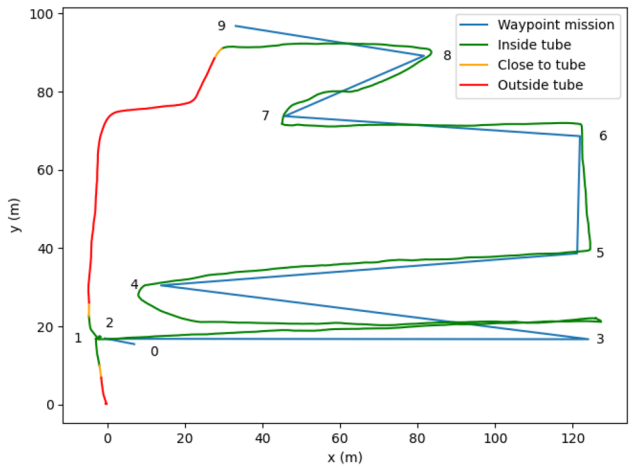


Fig. 6 Top-view of the tubing results. We remained within the tube (green) until the waypoint mission stopped at Waypoint 9. Then, the UA left the tube to fly to the home position

at *Waypoint 9* was never reset since the waypoint were not reached within the ϵ distance. Yet, all other waypoints were sufficiently reached, resetting the timer.

Geofencing of the operation area as well as the pre-planned flight trajectory allows to reduce the likelihood of unforeseen features a neural network might encounter. It is

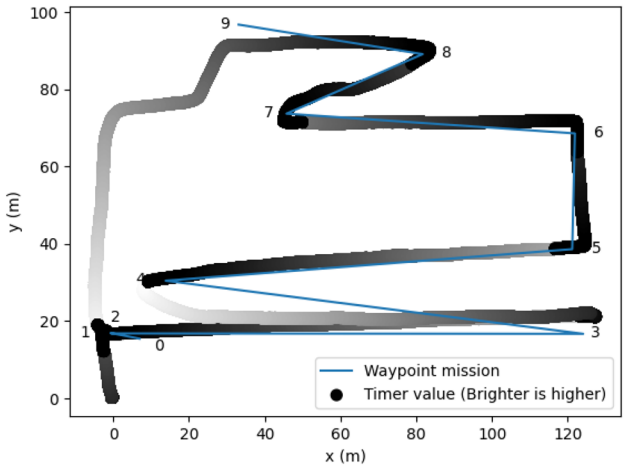


Fig. 7 Top-view of the tubing timer. At each waypoint, a timer is started that is reset when the next waypoint is reached within an ϵ distance. The brighter colours of the flight show that we did not reach Waypoint 4 in time and Waypoint 9 was outside the ϵ distance to reset the timer. All other waypoints reset the timer in time

one means of many to assure that the input distribution of a neural network is similar to the one during training.

4.3 OOD monitor

The images that are being fed into the neural network could suffer from multiple negative effects such as low contrast, blur, under- and overexposure or noise from a damaged camera sensor. Depending on the intensity of these effects, a reliable detection might become impossible. To filter out images with such effects the OOD monitor reads the current image frame from the connected camera and computes several properties of the frame. These properties include the brightness, the saturation, the entropy and the amount of edges in the frame. To derive the set of image properties, an interview with experts on environment perception was conducted. The computation of these properties is as follows.

Measuring the brightness of an image could be used to check if under- or overexposure exist in an image. It can be computed by converting the image to grayscale and calculating the mean value of all pixels. To get a value range between 0 and 1 the mean value is divided by 255. A very low value could indicate underexposure while a high value would indicate overexposure.

An image with low saturation could be an indicator for an reduced set of distinct features in an image, e.g. in case of hazy weather. To compute the saturation, the image is converted to the Hue Saturation Value (HSV) colour space before the mean value of the saturation channel is calculated. Again the saturation is normalized by dividing by 255.

Low contrast could also have a negative impact on the detection performance of the neural network. As there is no standardized method to compute the contrast of an image [50], we use the entropy instead as a low contrast also results in a low entropy measure [51]. The first step for the entropy computation is a conversion to grayscale. After that, the histogram of the image is computed. The probability of each pixel value can be determined by dividing the number of pixels with the same value by the total amount of pixels. With these probabilities we can compute the entropy with Eq. 1 where H corresponds to the entropy and $p(i)$ to the probability of one pixel value i.

$$H = -\sum_{i=0}^{255} p(i) \cdot \log_2 p(i) \tag{1}$$

To normalize the entropy, its value is divided by the maximum possible entropy H_{max} which corresponds to $H_{max} = -\log_2 \frac{1}{256}$.

Similar to the other metrics, the amount of edges in a frame is determined by first converting the image to grayscale. After that, a Laplace filter with a kernel size of 3 is applied to the grayscale image. Then the histogram of the filtered image is computed. The bright pixel values in the histogram correspond to edges while the dark pixels are areas without edges. We defined that all pixel values below 25 in the histogram are areas without edges while everything else corresponds to an edge. So the edge value would be 0 if all values in the histogram are below 25 or 1 if all values in the histogram would be above 25.

In addition to these general thoughts on image parameters, it is helpful to look at the characteristics of trained images. Specifically, we take a look at the above training distribution for the image parameters. Per image in the training data, a scalar value is calculated for each image parameter. This results in a training distribution for these parameters. We currently analyze this distribution in two steps: first, we determine a matching beta distribution and second we determine a quantile threshold where not enough data is currently present for sufficient trust in the ML network. However, this approach is one-dimensional. Therefore, if assessing multiple parameters and therefore dimensions, another step is required to handle the multiple dimensions. One example would be to declare the image as OOD if just one of the parameters is OOD. To directly assess OOD for multiple dimensions, we utilize the Mahalanobis distance metric. In contrast to Euclidean distance, the Mahalanobis distance takes into account the correlation between the different dimensions. A cut of distance is determined, where not enough data points are available to trust the ML network. The cutoff value is a trade-off between ML performance and availability of the ML network. The ML performance should increase with more images filtered out of the datastream. However, if a lot of images are filtered out, the present

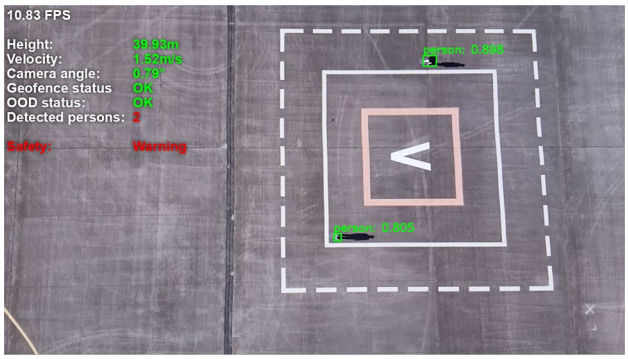


Fig. 8 Software demonstrator prototype of the Safe Operation Monitor (SOM), showing a screenshot of streaming an image from the onboard camera

detection is not available. Therefore, the determination of the cutoff value has an important impact.

4.4 Safety switch

Combined with the results from the ODD monitor and the OOD monitor, the bounding boxes from the ML algorithm are fed into a safety switch, which determines whether the results generated by the ML algorithm are trustworthy or not by combining the results of both monitors. Only when the monitors indicate that the ODD has not been violated and specify that an input image is in distribution, the results are considered to be trustworthy. The bounding boxes and the monitoring results are then drawn onto the frame. Afterwards, this annotated frame is streamed to the payload GCS where it can be analyzed by a payload operator. The visualization of such an annotated frame is depicted in Fig. 8. The annotations include information about the operational conditions such as the current altitude, velocity, camera angle and the geofence status. Additionally, the number of detected persons and potential out of distribution samples are displayed as well. The frames with annotations are then processed by a streaming component which converts them to an h.264 video stream that is transmitted to the payload GCS via UDP. The conversion and the streaming have both been implemented with the GStreamer framework.[3] To avoid a heavy CPU usage for the h.264 conversion, the built-in hardware encoding of the Nvidia Jetson Orin AGX is utilized. This enables an efficient video stream between the companion computer and the payload GCS.

[3] https://gstreamer.freedesktop.org/

5 Flight test

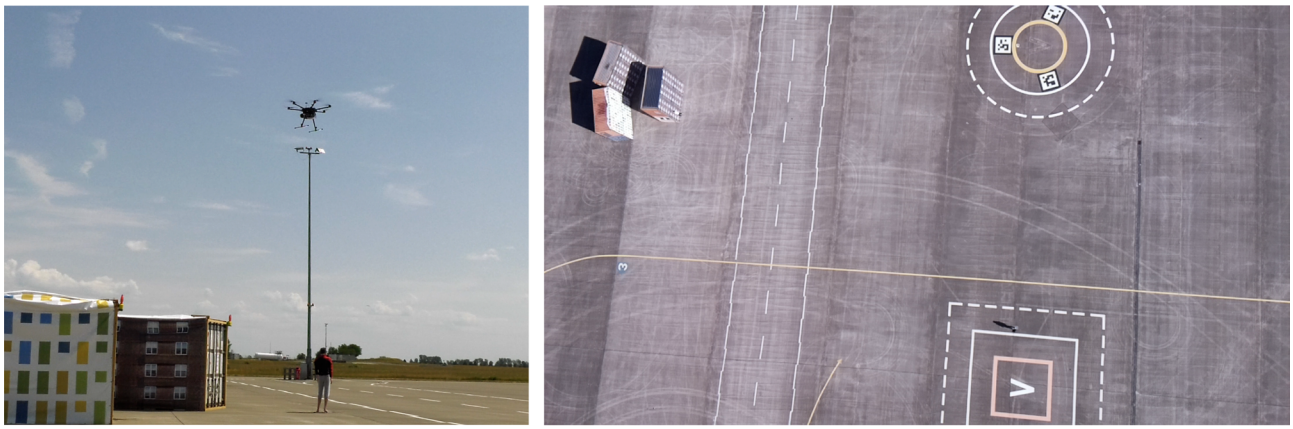
The objective of the flight test was to generate sets of images that are within and outside the ODD as well as to investigate the impact of ODD filtering. Note that, our monitoring architecture focuses on the performance impact of filtering ML output, while excluding hardware guidelines important for certification such as having independent hardware components for ML and monitoring. Relevant operational parameters for the flight test are altitude, velocity, camera angle, and image metadata such as brightness or entropy. The flight campaign was held on the first week of October 2022 at the *National Experimental Test Center of Unmanned Aircraft Systems* near Cochstedt. The test area includes two vertiports and a container cluster of six units, which mimic buildings (see Fig. 9). The humans to be detected were represented by mannequins which look very similar to real humans on aerial images. Using the mannequins alleviated the need to fly over humans and thus allowed for safer and more flexible flight testing. The mannequins are attached to a base plate that prevents them from falling over. Due to its grey colour, the base plate blends in with the concrete surface at the airport and is barely visible on aerial images. In total, twelve flights have been completed during the flight tests, see Table 1. All flights were conducted with the same waypoint coordinates. However, the altitude, flight speed, camera angle and positioning of the mannequins have been changed between flights. Additionally, three layers of transparent adhesive tape have been added onto the camera lens to simulate a blur effect during the last flight. Across all flights 6993 images have been recorded with the onboard camera.

5.1 Unmanned aircraft

For these flight tests, the City-ATM Demonstrator (CDO) hexacopter (Fig. 10) was used as the payload carrier for this new flight campaign. Originally built for City-ATM project [52] it serves as a versatile and reliable vehicle for the flight testing. For more detailed description of CDO please refer to our previous paper [5].

5.2 Modifications to unmanned aircraft

The payload was modified with new hardware components. Major modifications were made to the companion computer. The Raspberry Pi 4 and Intel NCS2 were replaced by the NVIDIA Jetson Orin. This boosted the onboard processing significantly through the increase in computational power. The previously used Raspberry Pi Camera Module 2 was changed to an industrial camera, with better optics and internal sensors. The camera chosen was the JAI-GO-2400 M-USB. The data link for communication



(a) Container cluster (b) Top view of vertiports and container cluster

Fig. 9 On the left (**a**), the container cluster surrounded with mannequins and UA in mid air. On the right (**b**), top view of the vertiports and the container cluster

Table 1 Table of performed flight tests with variations in ODD parameters, altitude, velocity and camera angle

	Altitude	Velocity	Camera angle	Mannequins	Notes
1st flight	40 m	5 m/s	Vertical	None	
2nd flight	40 m	5 m/s	Vertical	Position 1	
3rd flight	20 m	5 m/s	Vertical	Position 1	Route to last waypoint at altitude of 30 m
4th flight	20 m	15 m/s	Vertical	Position 1	
5th flight	80 m	5 m/s	Vertical	Position 1	
6th flight	10 m	5 m/s	Vertical	Position 1	
7th flight	40 m	5 m/s	Vertical	Position 2	Aborted after 2 min due to power issues
8th flight	40 m	5 m/s	Vertical	Position 2	
9th flight	40 m	5 m/s	45° tilted forward	Position 2	
10th flight	40 m	5 m/s	45° tilted backward	Position 2	
11th flight	15 m	15 m/s	Vertical	Position 2	
12th flight	40 m	5 m/s	Vertical	Position 2	Adhesive tape for blurr effect

Additionally, the first flight was performed without mannequins, and the position of the mannequins was switched with the 7th flight and following flights

Fig. 10 The UA CDO 002 Hexacopter

with the payload and the Pixhawk were unchanged from the previous flight campaign. Because of the bigger dimensions of the Jetson in comparison to a Raspberry Pi, the Payload rail was redesigned to fit all the components for the flight test under the UA. Additional peripheral devices such as a network switch and Battery Eliminator Circuit (BEC) were also integrated. The payload is completely powered by the UA's battery and protected against overcurrents through a fuse box. The modified placement of key components is shown in Fig. 11. The LiDAR seen mounted on the front of the UA in the pictures is used for another project.

A non-stable version of the PX4 software stack on the Pixhawk was needed to enable the MicroRTPS Messaging described in Sect. 4.2.2. However, this posed a safety risk,

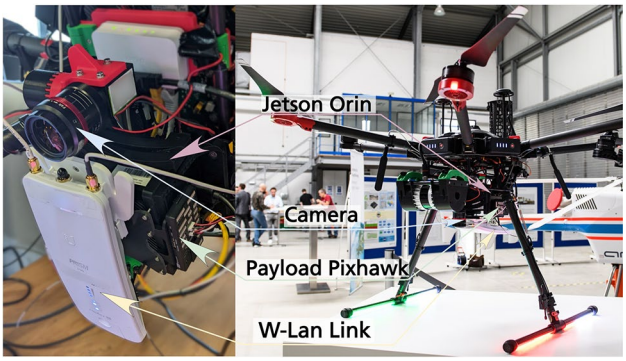


Fig. 11 The UA after full integration of the components

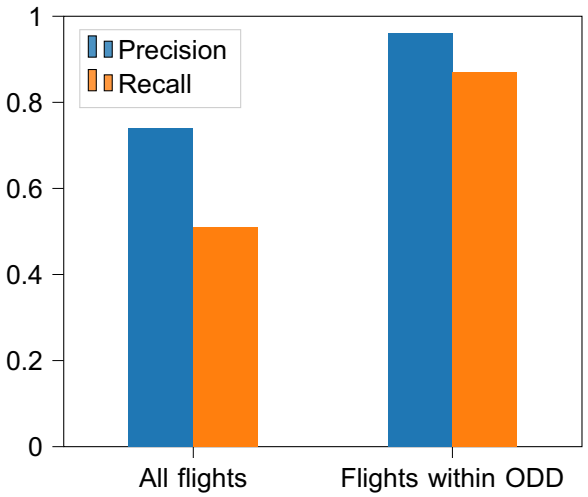


Fig. 12 Comparison of recall and precision between all flights and all flights within the ODD

as the Pixhawk might fail in mid air, resulting in a loss of the UA and the payload. Therefore, we used two concurrent Pixhawks for the flight tests. One Pixhawk for the actual flight controls and another Pixhawk for the payload as described in Sect. 4.2. Similarly, two separate GCSs have been utilized for the flight tests. The first GCS was used by the UA operator to supervise the individual flight missions, while the second GCS allowed the payload operator to monitor the performance of the onboard ML algorithm.

6 Results

A total amount of 6993 images has been recorded across flights. These images have been used to evaluate the overall software setup. First, some of the recorded images have been used to evaluate the performance of the ML component and the influence of the ODD on the detection performance. Second, the properties of the recorded images have been analyzed to determine if filtering of blurred images using the input monitor could be feasible.

6.1 Evaluation of ODD monitoring

To evaluate the performance of the ML algorithm, 587 images were randomly selected from the recorded images as a test set. The 587 images in the test set have been labelled manually and have been used to compute the precision and recall scores of the ML component for the flight tests. The precision P and the recall R are defined as $P = \frac{TP}{TP+FP}$ and $R = \frac{TP}{TP+FN}$ respectively, where TP corresponds to the number of true positives, FP to the number of false positives and FN to the number of false negatives.

In the context of object detection, a true positive exists when the bounding box of a detected object is similar to the bounding box of a ground truth object to a certain degree. A metric that evaluates the similarity of two bounding boxes is the Intersection over Union (IoU) which is defined as $IoU = \frac{A \cap B}{A \cup B}$ with A as the first bounding box and B as the second bounding box. Typically, an IoU threshold is used to determine whether two bounding boxes match each other. In this paper, a threshold of 0.5 is being used, which means that two bounding boxes are evaluated as true positive when their IoU score is larger or equal to 0.5. False positives on the other hand are bounding boxes that have been detected by the object detection algorithm, but can not be matched with a ground truth bounding box. In contrast, false negatives are the amount of ground truth bounding boxes that have not been matched with any detected bounding box.

In total, the ground truth annotations contain 684 objects. The resulting precision of the object detection algorithm is 0.74 and the recall is 0.51, see Fig. 12. However, if the images with altitudes higher than 50 m or lower than 20 m as well as images with other camera angles and velocities higher than 10 ms^{-1} are ignored, only the images within the ODD remain. In that case, the precision increases to 0.96 and the recall increases to 0.87. In summary, a positive impact of filtering out images outside the ODD for the detection performance is clearly visible.

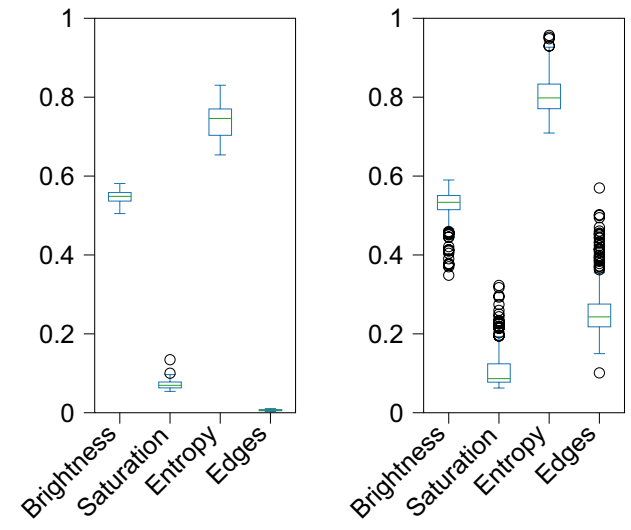
6.2 Evaluation of OOD monitoring

For their showcase of this methodology, a specific failure mode, a sensor error has been emulated. For the last flight, three layers of transparent adhesive tape have been used to simulate a blur effect. To check whether the blur can be detected using image properties, the recorded images have been analyzed using the input monitor described in Sect. 4.3.

These properties for the recorded images are displayed in two box plots which can be seen in Fig. 13. The first box plot refers to the images from the flight with the adhesive tape and the second box plot refers to the images from the other flights without the adhesive tape. In both plots the distributions of the brightness, the saturation and the entropy are fairly similar. Therefore these properties would not be suitable to distinguish between normal images and blurred images. However, the edges property differs significantly between the two plots. The highest edges value for the images with blur is 0.01 while 0.1 is the lowest recorded value for the remaining images. This demonstrates that a detection of blur in images using information about the amount of edges in an image might be feasible. This can safeguard the ML component and significantly improve the overall performance of the ML component.

6.3 Summary and limitations

From the results it shows that there is a significant difference in precision and recall over all images compared to the precision and recall over the images that are specifically inside the ODD. Therefore, the object detection algorithm has better results for images that comply with the ODD compared to images that violate the ODD. By monitoring the ODD during the operation, we safeguard the ML component against input images that would not result in an adequate performance and for which the ML algorithm is not intended for, such as discussed within the current EASA guidelines, see also Sect. 7.1. In addition the OOD monitoring was utilized to assess the data from the flight test demonstration. The analysis shows that there is a specific distribution from the gathered image data. The simulated sense of failure could be easily detected via OOD analysis. For more details on the OOD monitoring, a master thesis is available [53].



(a) 12th flight with blur effect. (b) Flights within ODD

Fig. 13 Image properties of flight test data

7 AI certification considerations

For the use case, selected objectives will be analyzed and discussed from the EASA guidance document [7, 8]. A special focus will be on OD, ODD, and OOD.

7.1 AI trustworthiness analysis

This building block ensures that there is an adequate high-level view on the ML component of the aircraft. It requires a characterization of AI, a safety assessment, a information security assessment, and an ethics-based assessment. In the following subsections the breakdowns and compliance rationale for these objectives is detailed.

7.1.1 Objective CO-04

*"CO-04: The applicant should define and document the ConOps for the AI-based system, including the task allocation pattern between the end user(s) and the AI-based system. A focus should be put on the definition of the OD and on the capture of specific operational limitations and assumptions."*5 [8]

This objectives requires the development of a ConOps document for the system. The objective states that a focus should be made on the definition of the OD and specific operational limitations and assumptions. For our use case, we define the OD to the daylight conditions, good weather conditions at the airfield of the Cochstedt airport. Additional details on the definition of OD and ODD are described in Sect. 3.

7.1.2 Objective CL-01

"CL-01: The applicant should classify the AI-based system, based on the levels presented in Table 2, with adequate justifications." [8]

This objective is for determining the AI level of the system, according to the EASA guidelines. The use case would be classified as level 1B AI: Automation support to decision-making. The human would look at the camera image and inspect the image and also the output of the ML constituent and detected bounding boxes of persons. However, it should be noted that the AI level is actually a question of system design. The human detection can be used to alert the pilot, it can be used to automatically abort the landing approach and utilize an alternative or emergency landing site. This decision could be designed to be overridable or non overridable

by the pilot. In our use case design the pilot gets informed and can then act as a fallback layer.

7.1.3 Objective SA-01

"SA-01: The applicant should perform a safety (support) assessment for all AI-based (sub)systems, identifying and addressing specificities introduced by AI/ML usage." [8]

For this objective a safety (support) assessment for all AI-based (sub)system should be performed. In addition to this, the objectives on safety risk mitigation also require this functionality to be evaluated as part of the safety assessment. While a complete safety assessment is out of the scope of this paper, some considerations of design time assurance are discussed here. Some thoughts on the safety risk mitigation during runtime, especially through operational design domain monitoring, are discussed in Sect. 7.4. The architectural mitigations at the system level to reduce the criticality of the AI-based components already at design time are discussed briefly in Sect. 7.3.1.

During the design process, the safety assessment required is essentially performed at a system level through a functional hazard analysis. Furthermore, the utilization of our human detection can have an impact on the safety of the operation. The risks arising from the introduction of the AI-based system to detect humans is assessed at the Concept of Operations level through a specific operational risk assessment . Particularly, the assumptions of controlled ground area in such an operation to reduce the number of uninvolved people on the ground are affected. E.g. if the AI-based system fails to detect the human, then the controlled ground area would be compromised.

In our context, the following two failures would be possible: a false positive detection of a human and a false negative detection of a human. A false positive detection would mean that the system would not be able to land on the vertiport automatically. This would not immediately result in an unsafe situation, as long as there is a feasible contingency procedure for this situation. Possible solutions would include landing on an alternative landing site, requesting human support from a pilot or remote pilot, waiting in a safe hover position and re-evaluating the situation after some time. However, it should be noted that a false detection might also lead to an unsafe situation, if there is no adequate mitigation or the mitigation is exhausted. For example, a false negative detection could also be an attack scenario and can effectively resemble a denial of service attack for a high rate of such occurrences, or at the same time in multiple locations (we thank our reviewers for pointing this out). A false-negative detection would mean that the system would execute an automated landing, although a human is in the vicinity. This would immediately result in an unsafe situation. Therefore, to increased safety in both cases, it is necessary to monitor the input and detect if the data are outside the ODD.

7.2 AI assurance

This building block of AI assurance is about the learning assurance as well as the explainability aspects during development and also post operation.

7.2.1 Objective DA-02

"DA-02: Based on (sub)system requirements that have been allocated to the AI/ML constituent, the applicant should capture the following minimum requirements for the AI/ML constituent:

- *safety requirements allocated to the AI/ML constituent;*
- *information security requirements allocated to the AI/ML constituent;*
- *functional requirements allocated to the AI/ML constituent;*
- *operational requirements allocated to the AI/ML constituent, including AI/ML constituent ODD monitoring and performance monitoring, detection of OoD input data and data-recording requirements;*
- *other non-functional requirements allocated to the AI/ML constituent; and interface requirements."* [8]

This objective discusses the requirements documentation and also underlines the importance of ODD and also OOD monitoring by establishing this as specific operational requirements for the AI constituents. Although important in the context of ODD, this is not in the scope of this paper.

7.2.2 Objective DA-03

"DA-03: The applicant should define the set of parameters pertaining to the AI/ML constituent ODD, and trace them to the corresponding parameters pertaining to the OD when applicable." [8]

This objectives requires the definition of the set of ODD parameters of the AI constituent. Furthermore, ODD parameters should be traced to the corresponding parameters of the OD, when applicable. The discussion of OD/ODD parameters for our use case is described in Sect. 3.

7.3 Human factors for AI

The building block of Human factors for AI is about operational explainability, human AI teaming, and modality of interaction.

7.3.1 Objective EXP-05, EXP-06, EXP-07, EXP-09

"EXP-05: The applicant should design the AI-based system with the ability to monitor that its inputs are within the specified operational boundaries (both in terms of input parameter range and distribution) in which the AI/ML constituent performance is guaranteed" [8]

"EXP-06: The applicant should design the AI-based system with the ability to monitor that its outputs are within the specified operational performance boundaries" [8]

"EXP-07: The applicant should design the AI-based system with the ability to monitor that the AI/ML constituent outputs (per Objective EXP-04) are within the specified operational level of confidence." [8]

"EXP-09: The applicant should provide the means to record operational data that is necessary to explain, post operations, the behaviour of the AI-based system and its interactions with the end user, as well as the means to retrieve this data." [8]

These objectives on monitoring inputs and outputs may also be fulfilled by the use of runtime assurance principles as described in Sect. 2.5. However, neither the ASTM F3269-21 [37] nor the accompanying explanatory article by Nagarajan et al [38] go into specific detail on the design of the warning and mitigation boundaries other than generic guidance on safety monitor switching thresholds. The responsibility lies on the user to interpret where these thresholds lie for their systems, and there are no requirements on when or how to switch from the complex function to the recovery function as this is considered to be implementation-specific. Schierman et al [42] provide a formal definition for three safety regions, i.e. Type I/II/III Safety Regions, which is more useful for designing warning and switching boundaries, see Fig. 14.

7.3.2 Objective EXP-19

"EXP-19: Information concerning unsafe AI-based system operating conditions should be provided to the end user to enable them to take appropriate corrective action in a timely manner." [8]

Since the safety properties of the system are clearly defined in each of these regions, a designer could adapt them for their system based on the mitigation strategies available to them. Warning boundaries could also be implemented while accounting for the time taken by a human operator or the recovery function to trigger a contingency action in order to prevent an excursion into unsafe regions.

For the purpose of explainability of the ML component, the pilot should be aware if the current system state is currently inside or outside of ODD parameters. Furthermore, the pilot should be trained to handle situations of exiting the ODD. Basically, as soon as the system is leaving the ODD,

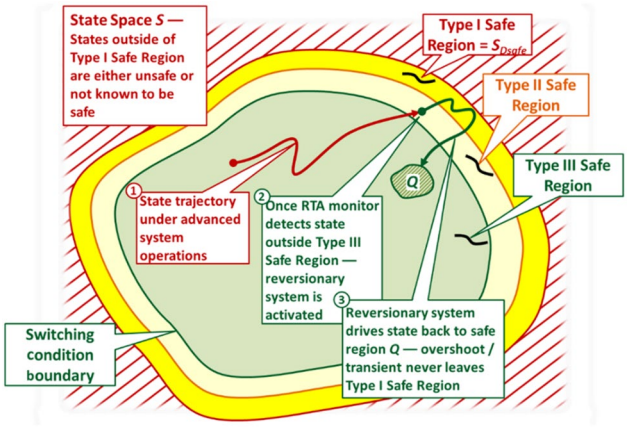


Fig. 14 Type I, II and III safety regions for run time assurance as defined in [42]

the ML functionality can no longer be trusted. However, it is possible to include multiple boundaries or warning levels before exiting the ODD completely. This would give the pilot additional time as well as information on how to handle the current situation and thus improve situational awareness.

The ODD monitoring will result in images being filtered from the datastream of the ML model. In these cases, a warning light will light up. Furthermore, a counter will count the overall ODD images as well as the ODD images with in the last 30 s. With this information, the pilot can determine further actions and possibly decide to abort the flight if necessary. The pilot will always see the image so that he himself can assess if there is a person in the image.

7.4 Safety risk mitigation

The last building block is the AI safety risk mitigation.

7.4.1 Objective SRM-01

"SRM-01: Once activities associated with all other building blocks are defined, the applicant should determine whether the coverage of the objectives associated with the explainability and learning assurance building blocks is sufficient or if an additional dedicated layer of protection, called hereafter safety risk mitigation, would be necessary to mitigate the residual risks to an acceptable level." [8] *"SRM-02: The applicant should establish safety risk mitigation means as identified in Objective SRM-01"* [8]

This objective states that an analysis of the coverage of objectives from the building block of explainability and learning assurance should be performed to assess if there are remaining risks that would need to be mitigated. In this work, the ODD aspects are focussed on. The coverage of explainability and learning assurance objectives cannot be

analyzed in the scope of this paper. However, the guidance states that with higher AI-level and higher criticality of the AI constituent the likelihood that SRM will be needed increases. Therefore, SRM is an essential part of the safety assurance strategy. In particular, for our use case we identified SRM to be required for the monitoring of the OD of the operation, the ODD of the ML model, specifically the quality of the image and training distribution of image parameters.

7.4.2 Objective SRM-02

"SRM-02: The applicant should establish safety risk mitigation means as identified in Objective SRM-01" [x] This objective describes the actual utilization of runtime monitoring and runtime assurance for the purpose of SRM. It is very important that runtime monitoring and runtime assurance are recognized as concepts for the mitigation of safety risks. The details of this will be discussed in the next subsection.

7.4.3 Anticipated MOC SRM-02-1

The RTA architecture as discussed in Sect. 2.5 can be used to fulfil the requirements in the anticipated MOC SRM-02-01. In the context of this reference architecture, the AI/ML constituent is the complex function, backed up by the traditional system (e.g. safety net) as a recovery function. Here, the requirement says to passivate the AI/ML constituent. Furthermore a recovery system is required. The question is, if the AI/ML constituent could be reactivated after recovery, or if the complex function is only allowed to have erroneous outputs once, i.e. as soon as the function is detected to be out of its operational design domain. If the AI/ML constituent is essential for mission performance, this would result in a low availability for the system if it receives inputs outside of the ODD frequently. This imposes implicit requirements on the user to build a more reliable AI/ML constituent or have a suitable safety net that can perform the same function to nearly the same standards. Otherwise it would be necessary to reduce the allowed operational scope for which the AI/ML constituent is allowed, in order to prevent frequent erroneous outputs due to inputs out of the ODD. In [38], the authors discuss the switching between complex and recovery functions with an emphasis on consideration of stable switching and chattering aspects to avoid frequently switching between the two functions. Graceful degradation of functionality is recommended as a best practice to prevent large instantaneous changes in performance (e.g. loss of function without prior warning of degrading performance).

In this research, the monitoring is done on additional sensor data, such as altitude, see Sect. 4.2.3, GNSS geofence as well as GPS tunnel, see Sect. 4.2.4. Therefore, there is a redundancy in sensors. In case of any geofence/ODD violation, control would be escalated to a human pilot as a fallback/recovery system. Regarding the AI use case, the image is being analyzed for person detection, see Sect. 4.3. A possible mitigation in this case would be to passivate any output from the ML function. The automated landing would be disabled completely and control would again be escalated to a human pilot as a fallback/recovery system.

The third requirement of the anticipated MOC above prescribes the evaluation of the SRM functions as part of a safety assessment and the need for establishing independence between the different SRM architectural mitigations. While Nagarajan et al [38] briefly discuss development assurance attributes in their work explaining the development of ASTM F3269, the concept of development assurance levels arising from a functional hazard assessment (FHA) is not explored in detail. This is a major gap in the ASTM standard practice, as there is a lack of guidance on how the architecture would align with current systems engineering guidelines such as SAE ARP 4754A in supporting a traditional certification process. Here, the reader is referred to the work by Peterson et al [54], who demonstrate the application of RTA in the context of a development assurance process for safety critical airborne applications with the example of a VTOL aircraft.

8 Conclusion and outlook

This work evaluated runtime monitoring techniques to enhance the safety and efficacy of ML components for UAM. Key focus areas included monitoring the ODD, OOD detection, and their integration into UAM scenarios. The use case in this case is the automated detection of persons in an onboard camera image. A total of 6993 images were collected across twelve flights, and these images were used to evaluate the systems performance under varied conditions. The results confirm that runtime monitoring plays an important role in ensuring operational safety by filtering inputs that fall outside the training distribution or designed parameters. This filtering process enhanced the reliability of the onboard ML algorithms. This can be used to safeguard the landing approach for an airtaxi.

The integration of runtime monitoring aligns with recent EASA guidance, particularly in leveraging the ODD framework for defining operational boundaries. This work demonstrates compliance with important objectives of safety and performance monitoring, while showcasing a real-world use case of person detection for air taxi landing. While full adherence to EASA guidelines was beyond the scope of this paper, select objectives were discussed in the context of ODD.

The integration of ML in the safety-critical domain of UAM is an ongoing effort. Standardization groups and

authorities are currently working hard to develop and establish new guidance for the safe integration of ML into aircraft. Based on the EASA guidance, all objectives that are related to ODD have been analyzed. This shows that the seemingly simple concept of ODD is utilized in all of the four building blocks of the EASA guidance. For selected objectives the compliance rationale as discussed in the context of our UAM use case. Additionally, the ODD for this use case is presented and discussed together with a brief discussion on safety considerations. However, not all of the related objectives could be discussed in detail.

Furthermore, the effects of monitoring exemplary ODD properties are evaluated. Although, achieving full compliance to EASA guidelines was outside the scope of this work, selected objectives could be analyzed, implemented and flight tested. Future work will build on this and deepen and/or broaden the discussion of the ODD concept to a larger extent. Further research and formalization of the concept of ODD can support the verification, safety assurance, and automation of future airtaxi operation. In the experimental setup of this work, the ODD monitoring and assurance are not yet integrated with the autopilot. Future work should aim to close the operational loop by integrating runtime monitoring outputs directly into automated decision-making systems.

In addition to that, it should be noted that the focus of this paper was on OD, ODD and OOD of AI systems. However, these concepts are not limited to AI systems. In fact, every complex system that cannot be sufficiently assured at design time can benefit from runtime monitoring and runtime assurance framework and from modelling on formalizing the operating conditions in this manner. Future work will bring these parts together into an overall framework as extension of our Operation Monitor.

Acknowledgements This work was partially supported by the Aviation Research Program LuFo of the German Federal Ministry for Economic Affairs and Energy as part of "Volocopter Sicherheits-Technologie zur robusten eVTOL Flugzustandsabsicherung durch formales Monitoring" (No. 20Q1963C). This paper is an updated and extended version of: Torens, Juenger, Schirmer, Schopferer, Zhukov, Dauer, Ensuring Safety of Machine Learning Components Using Operational Design Domain, AIAA SciTech Forum 2023, https://arc.aiaa.org/doi/10.2514/6.2023-1124 [5].

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability The raw data supporting the conclusions of this article are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors have no competing interests to declarethat are relevant to the content of this article.

References

1. Schuchardt, B. I., Becker, D., Becker, R.-G., End, A., Gerz, T., Meller, F., Metz, I. C., Niklaß, M., Pak, H., Shiva Prakasha, P., Schier-Morgenthal, S., Schweiger, K., Sülberg, J. D., Swaid, M., Torens, C., Zhu, C.: "Urban air mobility research at the DLR German aerospace center - getting the horizonUAM project started," in *AIAA Aviation and Aeronautics Forum and Exposition, AIAA AVIATION Forum 2021*, (2021). [Online]. Available: https://elib.dlr.de/143647/
2. Pak, H., Asmer, L., Kokus, P., Schuchardt, B. I., End, A., Meller, F., Schweiger, K., Torens, C., Barzantny, C., Becker, D., Ernst, J. M., Jäger, F., Laudien, T., Naeem, N., Papenfuß, A., Pertz, J., Shiva Prakasha, P., Ratei, P., Reimer, F., Sieb, P., Zhu, C., Abdellaoui, R., Becker, R.-G., Bertram, O., Devta, A., Gerz, T., Jaksche, R., König, A., Lenz, H., Metz, I. C., Naser, F., Schalk, L. M., Schier-Morgenthal, S., Stolz, M., Swaid, M., Volkert, A., Wendt, K.: "Can urban air mobility become reality? opportunities and challenges of UAM as innovative mode of transport and DLR contribution to ongoing research," *CEAS Aeronaut. J.*, (2024). [Online]. Available: https://elib.dlr.de/205131/
3. Schuchardt, B. I., Torens, C.: "Horizonuam - safety and security considerations," in *Optics2 Workshop Towards SAFE and SECURE Urban air mobility*, (2021). [Online]. Available: https://elib.dlr.de/143650/
4. Torens, C., Volkert, A., Becker, D., Gerbeth, D., Schalk, L. M., Crespillo, O. G., Zhu, C., Stelkens-Kobsch, T. H., Gehrke, T., Metz, I. C., Dauer, J.: "Horizonuam: Safety and security considerations for urban air mobility," in *AIAA Aviation and Aeronautics Forum and Exposition, AIAA AVIATION Forum 2021*, (2021). [Online]. Available: https://elib.dlr.de/143012/
5. Torens, C., Juenger, F., Schirmer, S., Schopferer, S., Zhukov, D., Dauer, J. C.: *Ensuring Safety of Machine Learning Components Using Operational Design Domain*. AIAA, (2023). [Online]. Available: https://doi.org/10.2514/6.2023-1124
6. Torens, C., Volkert, A., Becker, D., Gerbeth, D., Schalk, L., Crespillo, O. G., Zhu, C., Stelkens-Kobsch, T., Gehrke, T., Metz, I. C., Dauer, J.: *HorizonUAM: Safety and Security Considerations for Urban Air Mobility*. AIAA, (2021). [Online]. Available: https://doi.org/10.2514/6.2021-3199
7. EASA, "Concept Paper First Usable Guidance for Level 1 Machine Learning Applications". (2021). [Online]. Available: https://www.easa.europa.eu/en/newsroom-and-events/news/easa-releases-its-concept-paper-first-usable-guidance-level-1-machine-0
8. EASA, "EASA Concept Paper: guidance for Level 1 & 2 machine learning applications Issue 02," Mar. (2024). [Online]. Available: https://www.easa.europa.eu/en/document-library/general-publications/easa-artificial-intelligence-concept-paper-issue-2

9. Torens, C., Juenger, F., Schirmer, S., Schopferer, S., Maienschein, T. D., Dauer, J. C.: *Machine Learning Verification and Safety for Unmanned Aircraft - A Literature Study*. AIAA (2022). [Online]. Available: https://doi.org/10.2514/6.2022-1133
10. Torens, C., Durak, U., Dauer, J. C.: *Guidelines and Regulatory Framework for Machine Learning in Aviation*. AIAA (2022). [Online]. Available: https://doi.org/10.2514/6.2022-1132
11. EASA, "Artificial Intelligence Roadmap, A Human-Centric Approach to AI in Aviation, Version 1.0" (2020). [Online]. Available: https://www.easa.europa.eu/en/document-library/general-publications/easa-artificial-intelligence-roadmap-10
12. EASA, "Concepts of Design Assurance for Neural Networks (CoDANN)" (2020). [Online]. Available: https://www.easa.europa.eu/en/document-library/general-publications/concepts-design-assurance-neural-networks-codann
13. EASA, "Concepts of Design Assurance for Neural Networks (CoDANN) II" (2021). [Online]. Available: https://www.easa.europa.eu/document-library/general-publications/concepts-design-assurance-neural-networks-codann-ii
14. FAA, "Neural Network Based Runway Landing Guidance for General Aviation Autoland" Federal Aviation Agency (FAA) and Daedalean, Tech. Rep. (2021). [Online]. Available: https://doi.org/10.21949/1524481
15. SAE G-34, Artificial Intelligence in Aviation, "Artificial Intelligence in Aeronautical Systems: Statement of Concerns," SAE International, Tech. Rep. (2021). [Online]. Available: https://www.sae.org/standards/content/air6988/
16. Belcaid, M., Bonnafous, E., Crison, L., Faure, C., Jenn, E., Pagetti, C.: "Certified ml object detection for surveillance missions," (2024) [Online]. Available: arXiv:https://arxiv.org/abs/2406.12362
17. Dmitriev, K., Schumann, J., Holzapfel, F.: "Toward certification of machine-learning systems for low criticality airborne applications," in *40th Digital Avionics Systems Conference, DASC 2021 - Proceedings*, ser. AIAA/IEEE Digital Avionics Systems Conference - Proceedings. Institute of Electrical and Electronics Engineers Inc., 2021, publisher Copyright: 2021 IEEE.; 40th IEEE/AIAA Digital Avionics Systems Conference, DASC 2021 ; Conference date: 03-10-2021 Through 07-10-2021. [Online]. Available: https://doi.org/10.1109/DASC52595.2021.9594467
18. Dmitriev, Konstantin, Schumann, Johann, Holzapfel, Florian: *Toward Design Assurance of Machine-Learning Airborne Systems*. AIAA (2022). [Online]. Available: https://doi.org/10.2514/6.2022-1134
19. Dmitriev, K., Schumann, J., Holzapfel, F.: "Towards design assurance level c for machine-learning airborne applications," in *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, (2022), pp. 1–6. [Online]. Available: https://doi.org/10.1109/DASC55683.2022.9925741
20. Wasson, K. S., Voros, R.: "Deobfuscating machine learning assurance and approval," in *2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)*, pp. 1–10 (2024)
21. Torens, C., Durak, U., Nikodem, F., Dauer, J. C., Adolf, F.-M., Dittrich, J. S.: "Adapting Scenario Definition Language for Formalizing UAS Concept of Operations," in *AIAA Modeling and Simulation Technologies (MST) Conference*. Kissimmee, FL, USA: AIAA, pp. 1–8 (2018). [Online]. Available: https://doi.org/10.2514/6.2018-0127
22. Torens, C., Durak, U., Nikodem, F., Schirmer, S.: "Formally Bounding UAS Behavior to Concept of Operation with Operation-Specific Scenario Description Language," in *AIAA SciTech Forum - 55th AIAA Aerospace Sciences Meeting*. San Diego, California: AIAA, pp. 1–11 (2019). [Online]. Available: https://doi.org/10.2514/6.2019-1975
23. Stefani, T., Girija, A., Mut, R., Hallerbach, S., Krüger, T.: "From the Concept of Operations Towards an Operational Design Domain for Safe AI in Aviation," (2023). [Online]. Available: https://elib.dlr.de/197957/
24. SAE International, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. Surface Vehicle Recommended Practice J3016" (2016)
25. The British Standards Institution, Center for Connected and Autonomous Vehicles, "PAS 1883:2021 Operational Design Domain (ODD) Taxonomy for an Automated Driving System (ADS) ??? Specification," (2021). [Online]. Available: https://www.bsigroup.com/globalassets/localfiles/en-th/cav/bsi-cav-safety-benchmarking-report-2021-th.pdf
26. I. S. 33, "Road Vehicles – Test scenarios for automated driving systems – Specification for operational design domain," International Organization for Standardization Standard (2023)
27. Colwell, I.: "Runtime Restriction of the Operational Design Domain: A Safety Concept for Automated Vehicles," Master's thesis, UWSpace (2018). [Online]. Available: http://hdl.handle.net/10012/13398
28. Gyllenhammar, M., Johansson, R., Warg, F., Chen, D., Heyn, H.-M., Sanfridson, M., Söderberg, J., Thorsén, A., Ursing, S.: "Towards an Operational Design Domain That Supports the Safety Argumentation of an Automated Driving System," in *10th European Congress on Embedded Real Time Software and Systems (ERTS 2020)*, TOULOUSE, France (2020). [Online]. Available: https://hal.science/hal-02456077
29. Yu, W., Li, J., Peng, L.-M., Xiong, X., Yang, K., Wang, H.: Sotif risk mitigation based on unified odd monitoring for autonomous vehicles. J. Intell. Connect. Veh. **5**(3), 157–166 (2022). https://doi.org/10.1108/JICV-04-2022-0015
30. Koopman, P., Fratrik, F.: "How many operational design domains, objects, and events?" in *Workshop on Artificial Intelligence Safety 2019 co-located with the Thirty-Third AAAI Conference on Artificial Intelligence 2019 (AAAI-19), Honolulu, Hawaii, January 27, 2019*, ser. CEUR Workshop Proceedings, H. Espinoza, S. Ó. hÉigeartaigh, X. Huang, J. Hernández-Orallo, and M. Castillo-Effen, Eds., vol. 2301. CEUR-WS.org (2019). [Online]. Available: https://ceur-ws.org/Vol-2301/paper_6.pdf
31. Mehlhorn, M.A., Richter, A., Shardt, Y.A.: Ruling the operational boundaries: a survey on operational design domains of autonomous driving systems. IFAC-PapersOnLine **56**(2), 2202–2213 (2023)
32. Weissensteiner, P., Stettinger, G., Khastgir, S., Watzenig, D.: Operational design domain-driven coverage for the safety argumentation of automated vehicles. IEEE Access **11**, 12263–12284 (2023)
33. Kaakai, F., Adibhatla, S. S., Pai, G., Escorihuela, E.: "Data-centric operational design domain characterization for machine learning-based aeronautical products," in *Computer Safety, Reliability, and Security*, J. Guiochet, S. Tonetta, and F. Bitsch, Eds Cham: Springer Nature Switzerland, 227–242 (2023). [Online]. Available: https://doi.org/10.1007/978-3-031-40923-3_17
34. Lee, K., Lee, K., Lee, H., Shin, J.: "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," (2018). [Online]. Available: arXiv:https://arxiv.org/abs/1807.03888
35. Yang, J., Zhou, K., Li, Y., Liu, Z.: Generalized out-of-distribution detection: a survey. Int. J. Comput. Vision **132**(12), 5635–5662 (2024). https://doi.org/10.1007/s11263-024-02117-4
36. EASA and Daedalean, *Concepts of Design Assurance for Neural Networks (CoDANN) II with Appendix B*, European Union Aviation Safety Agency (EASA) and Daedalean (2024)
37. ASTM F38, "Standard Practice for Methods to Safely Bound Behavior of Aircraft Systems Containing Complex Functions

Using Run-Time Assurance" (2021). [Online]. Available: https://www.astm.org/f3269-21.html
38. Nagarajan, P., Kannan, S. K., Torens, C., Vukas, M. E., Wilber, G. F.: *ASTM F3269 - An Industry Standard on Run Time Assurance for Aircraft Systems* AIAA, (2021). [Online]. Available: https://doi.org/10.2514/6.2021-0525
39. EASA and Collins Aerospace, *Formal Methods use for Learning Assurance (ForMuLA)*, European Union Aviation Safety Agency (EASA) and Collins Aerospace (2023)
40. Hook, L. R., Clark, M., Sizoo, D., Skoog, M. A., Brady, J.: "Certification strategies using run-time safety assurance for part 23 autopilot systems," in *2016 IEEE Aerospace Conference*, 1–10 (2016). [Online]. Available: https://doi.org/10.1109/AERO.2016.7500817
41. Torens, C., Nikodem, F., Dauer, J.C., Schirmer, S., Dittrich, J.S.: Geofencing requirements for onboard safe operation monitoring. CEAS Aeronaut. J. **11**(3), 767–779 (2020). https://doi.org/10.1007/s13272-020-00451-0
42. Schierman, J.D., DeVore, M.D., Richards, N.D., Clark, M.A.: Runtime assurance for autonomous aerospace systems. J. Guid. Control. Dyn. **43**(12), 2205–2217 (2020). https://doi.org/10.2514/1.G004862
43. Skoog, M. A., Hook, L. R., Ryan, W.: "Leveraging astm industry standard f3269-17 for providing safe operations of a highly autonomous aircraft," in *2020 IEEE Aerospace Conference*, pp. 1–7 (2020). [Online]. Available: https://doi.org/10.1109/AERO47225.2020.9172434
44. Cofer, D., Amundson, I., Sattigeri, R., Passi, A., Boggs, C., Smith, E., Gilham, L., Byun, T., Rayadurgam, S.: "Run-time assurance for learning-based aircraft taxiing," in *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, pp. 1–9 (2020). [Online]. Available: https://doi.org/10.1109/DASC50938.2020.9256581
45. He, Y., Schumann, J.: "Statistical Analysis and Runtime Monitoring for an AI-based Autonomous Centerline Tracking System," *PHM Society Asia-Pacific Conference*, **4**, no. 1, (2023). [Online]. Available: https://doi.org/10.36001/phmap.2023.v4i1.3738
46. Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y. M.: "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7464–7475 (2023). [Online]. Available: https://doi.org/10.1109/CVPR52729.2023.00721
47. Božić-Štulić, D., Marušić, Ž, Gotovac, S.: Deep learning approach in aerial imagery for supporting land search and rescue missions. Int. J. Comput. Vision **127**(9), 1256–1278 (2019). https://doi.org/10.1007/s11263-019-01177-1
48. Bochkovskiy, A., Wang, C., Liao, H. M.: "Yolov4: Optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, (2020). [Online]. Available: https://doi.org/10.48550/arXiv.2004.10934
49. Baumeister, J., Finkbeiner, B., Kohn, F., Löhr, F., Manfredi, G., Schirmer, S., Torens, C.: Monitoring unmanned aircraft: specification, integration, and lessons-learned. In: Gurfinkel, A., Ganesh, V. (Eds.) Computer Aided Verification, pp. 207–218. Springer Nature Switzerland, Cham (2024)
50. Rizzi, A., Algeri, T., Medeghini, G., Marini, D.: "A proposal for contrast measure in digital images," in *Conference on Colour in Graphics, Imaging, and Vision*, vol. 2004, no. 1. Society for Imaging Science and Technology, 2004, pp. 187–192
51. Mello Román, J. C., Vázquez Noguera, J. L., Legal-Ayala, H., Pinto-Roa, D. P., Gomez-Guerrero, S., García Torres, M.: "Entropy and contrast enhancement of infrared thermal images using the multiscale top-hat transform," *Entropy*, **21**, no. 3, (2019). [Online]. Available: https://www.mdpi.com/1099-4300/21/3/244
52. Kern, S., Geister, D., Korn, B.: "City-atm – demonstration of traffic management in urban airspace in case of bridge inspection," in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, pp. 1–10 (2019). [Online]. Available: https://doi.org/10.1109/DASC43569.2019.9081663
53. Kardatzke, S.:"Laufzeitüberwachung neuronaler netze für die kameragestützte umgebungswahrnehmung autonomer drohnen," Hochschule Osnabrück, Tech. Rep (2023). [Online]. Available: https://elib.dlr.de/202952/
54. Peterson, E. M., DeVore, M., Cooper, J., Carr, G.: "Run Time Assurance as an Alternate Concept to Contemporary Development Assurance Processes," NASA, Tech. Rep. NF1676L-36112 (2020). [Online]. Available: https://ntrs.nasa.gov/citations/20200003114
55. EASA and Daedalean, Concepts of Design Assurance for Neural Networks (CoDANN) II with Appendix B, European Union Aviation Safety Agency (EASA) and Daedalean, Jan. 2024. [Online]. Available: https://www.easa.europa.eu/en/documentlibrary/general-publications/concepts-design-assurance-neuralnetworks-codann-ii
56. EASA, "Artificial Intelligence Roadmap, A Human-Centric Approach to AI in Aviation, Version 2.0" (2023). [Online]. Available: https://www.easa.europa.eu/en/document-library/general-publications/easa-artificial-intelligence-roadmap-20
57. EASA, "Easy access rules for unmanned aircraft systems (regulations (eu) 2019/947 and (eu) 2019/945)," 2021. [Online]. Available: https://www.easa.europa.eu/en/document-library/easyaccess-rules/easy-access-rules-unmanned-aircraft-systemsregulations-eu