IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 20, 2019, accepted August 25, 2019, date of publication August 29, 2019, date of current version September 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2938249

UAV-Based Situational Awareness System Using Deep Learning

RÚBEN GERALDES[1], ARTUR GONÇALVES[1], TIN LAI[2], MATHIAS VILLERABEL[3], WENLONG DENG[4], ANA SALTA[5], KOTARO NAKAYAMA[6], YUTAKA MATSUO[6], AND HELMUT PRENDINGER[1]

[1]National Institute of Informatics, Tokyo 101-8430, Japan
[2]Faculty of Engineering, The University of Sydney, Sydney, NSW 2006, Australia
[3]Department of Computer Science, Sorbonne University, 75005 Paris, France
[4]EPFL, 1015 Lausanne, Switzerland
[5]INESC-ID, 1000-029 Lisbon, Portugal
[6]Department of Technology Management and Innovation, Graduate School of Engineering, The University of Tokyo, Tokyo 113-8654, Japan

Corresponding author: Helmut Prendinger (helmut@nii.ac.jp)

This work was supported by the grant on "Research on improving predictability by blending deep learning and symbol processing" provided to the Graduate School of Engineering, The University of Tokyo under Grant Kakenhi 16H06562.

ABSTRACT Situational awareness by Unmanned Aerial Vehicles (UAVs) is important for many applications such as surveillance, search and rescue, and disaster response. In those applications, detecting and locating people and recognizing their actions in near real-time can play a crucial role for preparing an effective response. However, there are currently three main limitations to perform this task efficiently. First, it is currently often not possible to access the live video feed from a UAV's camera due to limited bandwidth. Second, even if the video feed is available, monitoring and analyzing video over prolonged time is a tedious task for humans. Third, it is typically not possible to locate random people via their cellphones. Therefore, we developed the Person-Action-Locator (PAL), a novel UAV-based situational awareness system. The PAL system addresses the first issue by analyzing the video feed onboard the UAV, powered by a supercomputer-on-a-module. Specifically, as a support for human operators, the PAL system relies on Deep Learning models to automatically detect people and recognize their actions in near real-time. To address the third issue, we developed a Pixel2GPS converter that estimates the location of people from the video feed. The result – icons representing detected people labeled by their actions – is visualized on the map interface of the PAL system. The Deep Learning models were first tested in the lab and demonstrated promising results. The fully integrated PAL system was successfully tested in the field. We also performed another collection of surveillance data to complement the lab results.

INDEX TERMS Deep learning, object detection, action recognition, situational awareness, search and rescue, surveillance.

I. INTRODUCTION

Situational awareness by Unmanned Aerial Vehicles (UAVs), also known as 'drones', is important for many applications, such as search and rescue, surveillance, disaster response, and so on. In these applications, a core task is to locate people, animals or objects, and respond in a timely manner. In security applications, it is also beneficial to understand the type of action a person is performing.

Human resource in situational awareness is a key concern, as human performance is limited by fatigue and waning concentration when continuously watching a video feed. Here, an automated surveillance approach can support humans in analyzing the video feed and alert them promptly about detected persons in the area of interest, such as an area affected by disaster or big outdoor event. Such collaborative work is sometimes called "human-agent collectives" [1].

To improve situational awareness, we developed the PAL (Person-Action-Locator) system, which uses a visual camera mounted to a UAV and intelligent information processing to detect people, recognize their actions, and pin their location on a map shown on the PAL interface. The core component of the PAL system uses Deep Learning models that have been trained for multi-person detection and multi-action

The associate editor coordinating the review of this article and approving it for publication was Zhen Li.

recognition. Then, a dedicated converter translates pixels (representing a person) to a GPS location.

Unlike existing approaches for person detection, the perspective from the UAV's camera has a high variance of camera angles and often captures from a large distance to the target object. Therefore, we developed a Deep Learning model called "Position and Orientation Invariant Neural Network" (POINet), which is designed to analyze imagery from the UAV's view. It produces bounding boxes by first generating multiple areas of interest on the input image, which act as a segmentation mask to locate the rough position of people. Then, the corners of the boxes are anchored by generating bounding boxes that best fit the areas of interest. In this way, our approach overcomes the difficulty of learning a broad representation of humans from different camera angles. Furthermore, we developed a Deep Learning model called "Activity Network" (ActivityNet), which uses temporal information to predict persons' actions to increase the performance, compared to previous work [2].

Our PAL system aims to achieve near real-time person detection and action recognition.[1] Therefore, we needed to develop light-weight Deep Learning models, which can operate onboard the UAV.

The rest of this article is organized as follows. Section II first discusses the related work on several situational awareness topics, such as search and rescue, surveillance, data collection and disaster response. Then we report on other works on human detection and action recognition. Section III describes the architecture and main modules of our PAL system: Meta Camera, Deep Learning, Pixel2GPS converter, and User Interface. This section provides standalone results from our work on the Deep Learning models and the Pixel2GPS converter. Section IV reports on the results of field testing the fully integrated PAL system. Section V summarizes the paper and highlights its main contributions.

II. RELATED WORK

A. SITUATIONAL AWARENESS

The three main elements of situational awareness are [3]:

- Perception of the current situation;
- Comprehension of said situation; and
- Projection of the future condition.

In our work, we focus on the first element, automated perception of the current situation. In the following, we discuss some important works on situational awareness.

1) SEARCH AND RESCUE

Search and rescue relies mainly on situational awareness. If a person is missing, certain steps must be followed according to the context and probabilities related to the person's status and behaviors. Three main tasks were mentioned in [4]: investigation, containment and search. All of these tasks must be done as quickly as possible to prevent the increase of risk for the safety of the missing person. During the investigation task, useful information about the subject is collected, including their plans of activity before they got lost. This will help the investigators to understand what area has the highest probability of the person's whereabouts. The containment plan aims to prevent the search area from expanding. The more time has passed before the missing person is found, the bigger the search area becomes. Here, the deployment of UAVs can help to scan the search area quicker and possibly find the missing person sooner. It also reduces the risk to the search team's safety.

Nowadays, UAVs are already being used to help the search. Some of the approaches can detect people [5], using Deep Learning such as Convolutional Neural Network (CNN). An autonomous search and rescue system that incorporates CNN and a novel algorithm, Image-Based Visual Servoing (IBVS), which is based on Deep Deterministic Policy Gradients (DDPG), was developed in [6]. In contrast to these approaches, which focus on indoor applications, our system can be used on the field. Other works, such as [7], suggest methods to improve the search area coverage in the context of search and rescue.

2) SURVEILLANCE

Surveillance systems are currently used in diverse applications, ranging from private to military. Some have simple objectives, such as the detection of intruders, whereas others are more complex and need to identify potential threats.

In a private home, the installed system is required to detect the presence of an intruder. An intrusion detector can work mostly autonomously. In public places with high levels of safety risks, such as a crowded space during an event, detection is not sufficient. Instead, it is necessary to constantly monitor any suspicious behavior among the crowd. Here, it is also required to understand if there is any suspicious behavior. For the event case, understanding what is suspicious behavior is not trivial, and human expertise is required to make a decision.

Many surveillance systems already operate autonomously or semi-autonomously. However, most of these systems rely on fixed cameras with limitations in camera movement. Reference [8] present algorithms for cooperative multi-sensor surveillance to support a single human operator in monitoring the video feed. Here, real-time video processing transforms the video feed into descriptions of objects and events. The concept of multi-scale spatio-temporal tracking in smart video surveillance is explored in [9]. It aims to help the human operator in real-time threat detection and forensic investigation, based on background subtraction, salient motion detection for object detection, object classification, face cataloging and movement analysis. Other systems utilize indoor ground based robots with cameras [10] [11].

Unlike these works, our system extends these surveillance approaches to the outdoor area with the possibility of free camera movement. There are existing works that suggest

[1] Note, however, that real-time response is not required in our context, which is different from a "Detect and Avoid" situation, where the drone has to respond immediately to avoid other airborne hazards.

the use of UAVs for surveillance together with methods to improve mission planning [12], multiple UAVs control [13], as well as addressing the problem of occlusion from the camera's vision [14] [15].

3) DISASTER RESPONSE

In the aftermath of a disaster, there is an immediate need for emergency services. If it is not possible to find people in need of assistance, a search is required to find people that have been reported as missing. Additionally, UAVs can be deployed to scan the affected area for potential unknown victims.

These systems mainly depend on human-agent collectives where the agents work on collecting and filtering data as well as assisting in coordination and task allocation [1]. This work uses crowdsourcing to understand where potential victims or structure damage occurred and then deploys UAVs monitored by human operators to obtain video feeds of the locations. However, monitoring multiple video feeds requires several human operators as ''sensors''.

For that reason, automatic detection is preferable, even if some human monitoring is kept for redundancy. A planning approach for a disaster response system that searches for possible victims by detecting mobile phones to obtain probable locations of missing people is described in [16]. However, there might be cases where such kind of search is not possible, e.g. the phone is broken, or without battery, or the person is not carrying the phone.

For such situations, people detection via a UAV-based visual surveillance system, such as our PAL system, is an important and useful technology. Further, post-disaster mapping by UAVs is important for disaster response [17].

4) DATA COLLECTION

The use of UAVs for data collection has been increasing dramatically in recent years, as UAVs can perform noninvasive work, inspect places that are hard or dangerous to reach, and reduce the risk to human workers. For instance, in wildlife studies, the use of UAVs can reduce the consumption of resources and also reduce the risks to both animals and researchers. UAVs have been already used for automatic identification and data capture (AIDC) for specific cases. For example, [18] present a solution for 3D scene reconstruction of forest areas and [19] introduce an airborne multi-spectral imaging system for small UAVs for the purpose of data collection, more specifically for measuring vegetation.

B. HUMAN DETECTION AND ACTION RECOGNITION

1) TRADITIONAL COMPUTER VISION TECHNIQUES

Object and human detection have been studied extensively in the field of computer vision. Computer vision techniques involve transforming the raw pixel data into formats that allow a computer to extract useful information. Traditional methods for human detection include the use of histograms of oriented gradients (HOG) for object recognition [20], the use of geometry information to learn classifiers with a connected Riemannian manifold [21], and the utilization of depth cameras for building a 3D surface model for improved detection and tracking [22].

However, these approaches require computer vision experts to handcraft features that appear useful in some specific application. Further, the extracted representation of features is hard to generalize outside the specific domain. For such tasks, Deep Learning approaches have shown great success in recent years, and became the state-of-the-art approach.

2) DEEP LEARNING OBJECT DETECTION MODELS

Deep Learning uses a data-driven method for the neural network to learn discriminative features directly from raw pixels. For example, the model can jointly learn HOG features with deformation and occlusion handing [23], use multiple part detectors to combine as a strong pedestrian detector [24], and use CNN to detect objects of interest from UAV imagery [5]. These methods have demonstrated extraordinary results on visual problems such as detecting occluded objects, which is difficult to accomplish using traditional handcrafted HOG features.

The object detection task has also been framed as a regression problem by densely generating bounding boxes and spatially separate them with a neural network [25]. The unified architecture achieves high accuracy on class predictions. However, it relies on generating bounding boxes with a prior probability of box sizes and ratio.

Video feeds captured by UAVs are usually overhead shots with a wide variety of orientations. Therefore, current models perform poorly on datasets collected by UAVs. However, robust human detection is an essential part of our PAL system that aims to detect and locate people. We will describe our solution in the following section.

3) DEEP LEARNING ACTION RECOGNITION MODELS

Action understanding helps to provide context on the likely activities that the detected person is performing. It is also an essential aspect for a situational awareness system as it provides contextual and situational information for the captured scenes, which enables the system to gain more knowledge on the situation and be able address it properly.

Recent works for multi-person action recognition take a sequential approach by separating each component and optimize each part separately. A person is first detected by a CNN model and tracked with an algorithm, then a feature representation is extracted for each person to reason on their individual action [26]. This approach requires a large amount of processing time as it needs to repeat the process for each detected person; hence, it does not scale well on a UAV system.

A 3D CNN has also been proposed for action recognition [27]. It is based on the same concept as a regular CNN, but also performs convolutions in the temporal axis. It achieves high accuracy on action understanding with the cost of substantial computational time. Therefore, the limitation

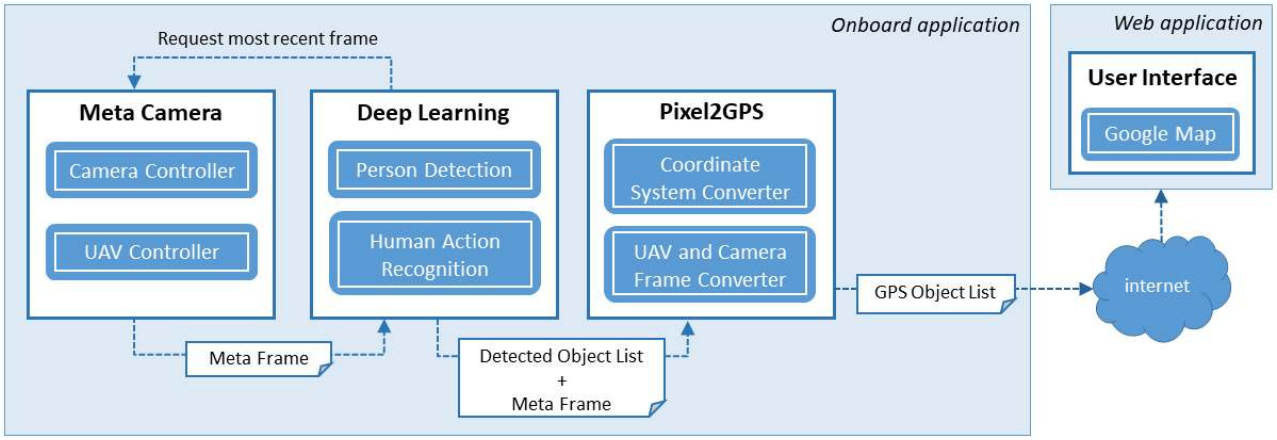


FIGURE 1. Person-action-locator (PAL) system architecture.

of computational power and battery power on current UAVs restrict the applicability of this technique to perform inferences onboard. We address these problems by developing a deep learning model that is designed to meet the onboard requirements.

III. THE PERSON-ACTION-LOCATOR (PAL) SYSTEM

The Person-Action-Locator (PAL) system can automatically detect people, recognize their actions and calculate their current GPS position onboard the UAV. This information is then displayed on a map user interface web application. The role of this system is to support the search process in search and rescue and surveillance scenarios. Our system is composed by an onboard application as the core system, and a web application, as shown in Fig. 1. The main modules are:

- **Meta Camera**: gathers video data together with UAV's telemetry data;
- **Deep Learning**: performs multi-person detection and action recognition;
- **Pixel2GPS**: converts a pixel coordinate into a GPS coordinate;
- **User Interface**: displays the result of the onboard application, such as the people's position and their respective actions, on a map;

We will now describe each of these modules in more detail.

A. META CAMERA

The Meta Camera is a module that captures an image from the UAV's onboard camera video feed stream and retrieves the telemetry of the UAV at the same time. The retrieved telemetry consists of the UAV's GPS position, height relative to the ground takeoff position, roll, pitch and yaw angles. This information is required by the Deep Learning and Pixel2GPS modules. The image is first fed to the Deep Learning module, and the corresponding telemetry is then used in the Pixel2GPS module as a reference to convert the pixel coordinates found by the Deep Learning module into GPS coordinates.

The Meta Camera is triggered to capture the most recent image and respective telemetry, every time the previous one has been fully processed by the rest of the system.

B. DEEP LEARNING

In our PAL system, Deep Learning (DL) is used for multi-person detection and multi-action recognition. We aim for lightweight DL models that allow us to execute inference onboard the UAV.

The most current image (or frame) requested to the Meta Camera is passed to the human detection component to produce a bounding box around each person. Then, the bounding boxes are used to crop the original image to produce a sequence of consecutive boxes of each detected person. The sequence of boxes is used as the input to the action recognition component, which predicts the probability that a person is performing some action of the action set.

Fig. 2 illustrates the information flow in our onboard system. Video feed frames captured from the UAV's camera are used as the input data for the model. The input is first passed to a DL model called "Position and Orientation Invariant Neural Network" (POINet), which extracts the point of interest and generates bounding boxes around the objects. The extracted bounding boxes are then passed to a second DL model called "Activity Network" (ActivityNet) to learn the persons' temporal information and predict their current actions.

Our Deep Learning models were trained and evaluated on the Okutama dataset [2]. It is a dataset of images captured by two different UAVs with varying camera angles and altitudes (15 m, 20 m, 25 m, and 30 m), taken during morning and evening periods. There is no restriction on altitude except the legal ceiling of 150 m. To train a Deep Learning model for higher altitudes, a new dataset would have to be collected.

The current dataset consists of a labelled bounding box around each person and one or more actions of that person. A person can perform one *primary* action, such as standing or walking, and one *secondary* action, such as carrying or pulling.

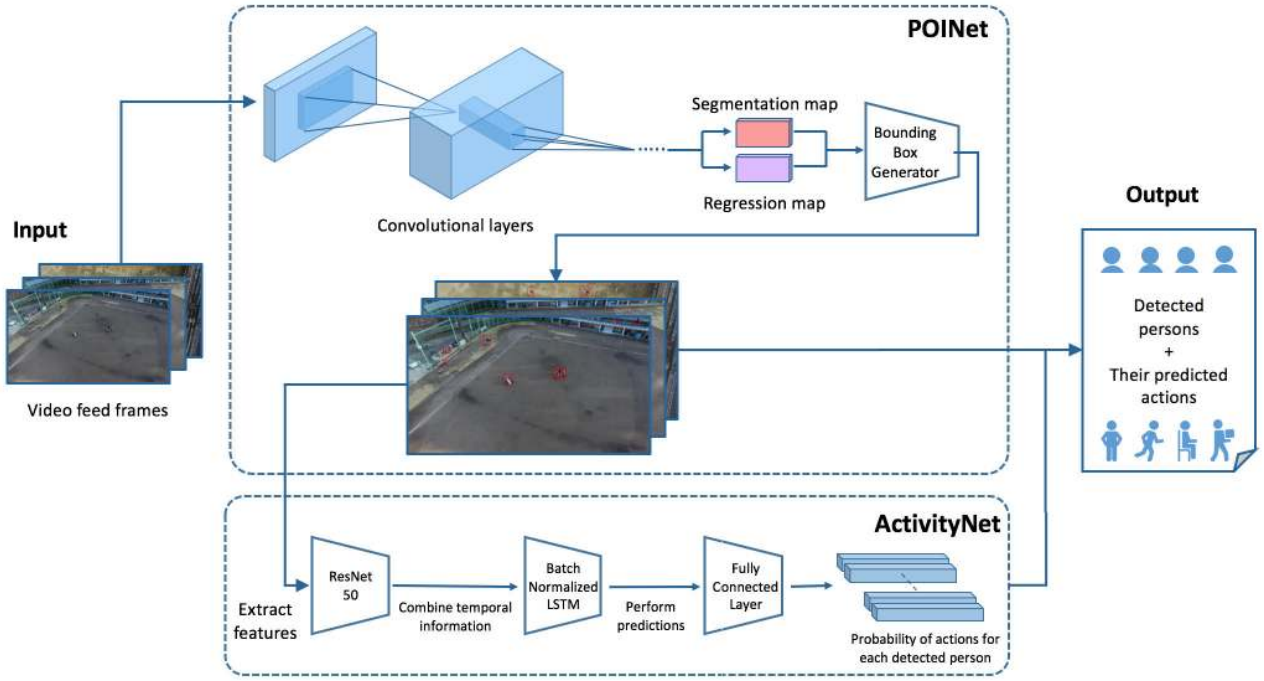
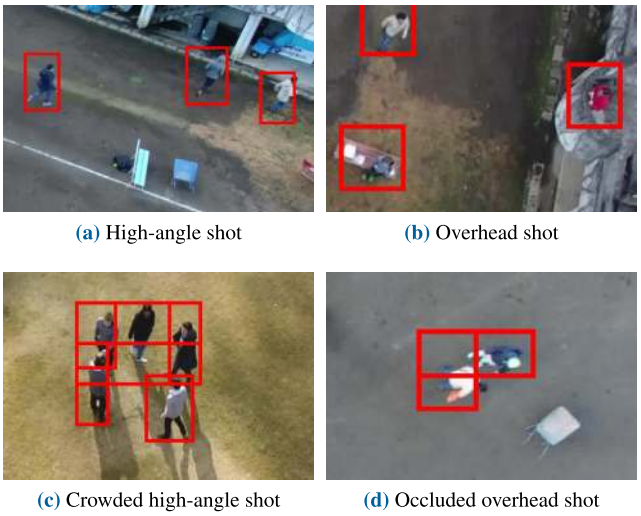


FIGURE 2. Architecture of the deep learning models for human detection and action prediction .



(a) High-angle shot (b) Overhead shot

(c) Crowded high-angle shot (d) Occluded overhead shot

FIGURE 3. Examples of different scenarios in the dataset.

1) DEEP LEARNING MODEL FOR MULTI-PERSON DETECTION

Video feeds from a UAV's camera often have a high variance on the camera angles, commonly ranging from high-angle to overhead shots, with most of the footage captured from a large distance from the targets, as shown in Fig. 3.

The nature of these camera angles limits the applicability of most current state-of-the-art object detection models. The reason is that most current models are designed to be trained on images that are taken from consumer cameras on ground level, or from a relatively horizontal camera angle. Some models also rely on prior beliefs on the aspect ratio of the bounding boxes [25]. This approach is reasonable if the orientation of the objects of interest has similar appearance across the frames.

Therefore, we developed POINet, which utilizes a deep convolutional neural network to extract features from the input images. The convolutional neural network [28] is made up of a sequence of layers that act as detection filters for the presence of specific patterns in the image. It can extract abstract features and learn to localize objects in the image space.

POINet utilises Mobilenetv2 [29] as its internal convolutional layers to extract multiple features in different scales. Then, POINet uses multiscale features to generate a segmentation map and regression map based on the original image. The segmentation map acts as a mask to establish the area of bounding boxes, while the regression localizes the corners of the bounding box. With the use of both maps, we are able to generate the coordinates of bounding boxes that overlay each of the people on the image. The output of POINet is illustrated in the middle of Fig. 2, where the red boxes are output bounding boxes generated by POINet. Some examples of the output are also shown in Fig. 3.

The model utilizes segmentation and regression map to generate bounding boxes directly in the image space, which overcomes the difficulty of high variation of camera angles among the input images. For an input size of $1280 \times 720 \times 3$ pixels, the number of parameters of our POINet is 10.5 million (M), which can be considered a lightweight design when compared to other approaches reported in [29]. Instead of using a much deeper CNN to learn the broad representation of persons, which is computationally

TABLE 1. Results for mAP of action recognition using different frame rate, in the test data of [2]. PA abbreviates "primary action" and SA abbreviates "secondary action".

FPS	PA	SA
0.667	20.8	11.5
3.0	22.5	11.0

expensive for UAVs, our approach achieves high accuracy with minimal impact on performance.

We evaluated our POINet model on the Okutama dataset. We have trained our model for more than 100 epochs until the training loss has converged.

Our model achieves a Recall of 86%, Precision of 85%, and F1 score of 84%. Compared to previous works [2], which uses Single Shot MultiBox Detector (SSD) [30] and has a F1 score of 72.3%, our model is able to more robustly detect persons regardless of the orientation of the camera angles.

2) DEEP LEARNING MODEL FOR MULTI-ACTION RECOGNITION

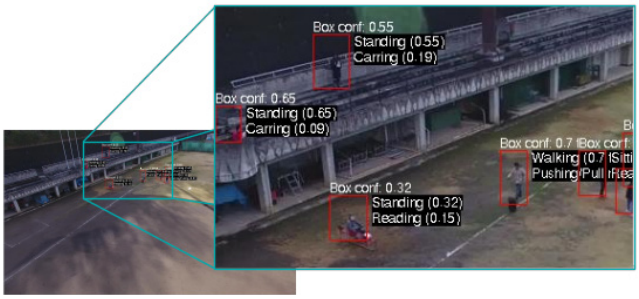
The Okutama dataset contains the following types of actions.

- Primary actions (non-interactive actions): Running, Walking, Lying, Sitting, Standing
- Secondary actions are comprised of
  - Human-to-human interactions: Handshaking, Hugging
  - Human-to-object interactions: Reading, Drinking, Pushing/Pulling, Carrying, Calling

ActivityNet uses a different approach to capture the features of the detected persons. Instead of purely relying on the spatial appearance of the persons, ActivityNet learns the temporal information using a modified Long Short-Term Memory (LSTM) network [31]. LSTM is a type of recurrent neural network that can learn from a sequence of data, such as video feeds, to produce informed predictions based on the changes in features over time. We extract the cropped person's features with a ResNet-50 [32], and pass the output to a Batch Normalized LSTM (BNLSTM) [33] to learn the person's action sequence. Using ResNet-50 features extractor we can induce an implicit tracking over time using the height and width of the bounding boxes. The output of BNLSTM are features that have incorporated information from previous frames, which enables the model to learn the features of time-series actions.

Finally, a fully connected layer is used to convert the features into a list of vectors that represent the probability of actions for each detected person. The output of the entire deep learning model is a list of bounding boxes, one for each detected person, and their corresponding predicted actions. The number of parameters of ActivityNet is 44.5 M, which achieves a lightweight design when compared to other works reported in [34].

Unlike ActivityNet, SSD cannot predict multiple actions. With ActivityNet, at 3 frames per second, we achieve a mean Average Precision (mAP) of 22.5% for primary actions



(a) Eye-level image prediction

(b) Overhead image prediction

FIGURE 4. Example output of the Deep Learning module.

(F1 score is 20%), and mAP of 11% for secondary actions (F1 score is 7%) (see Table 1). The accuracy of detecting primary actions is 80%, and 84% for secondary actions. For single action recognition, our model achieves mAP of 40%, which outperforms to SSD's mAP of 18.8% [2].

The model was trained using two NVIDIA GTX 1080 Ti and 32 GB of RAM. To test the performance of our model on our target hardware on the UAV, we transferred our Deep Learning model alongside the trained weights to an NVIDIA Jetson Tegra X2, which uses the Pascal GPU architecture with 256 CUDA cores and 8 GB of RAM. Since the onboard processing power of the Jetson Tegra X2 is limited, the model cannot process the video feed at a high frame rate. The inference process is able to perform, on average, in 1.5 seconds per frame, i.e. 0.667 frames per second (FPS). We used the test data from [2] and changed the frame rate to match it with the inference time of the onboard system. The comparison is reported in Table 1. With these results we can suppose that even with a lower frame rate the overall performance of the model is not affected significantly.

Action recognition uses "historical" information (e.g. past 4 frames of a person). Therefore, the speed of the UAV should not be higher than 12.5 m/s, for an altitude of 30 m and camera angle of 45 degrees, to ensure the person remains in the field of view of the UAV.

Fig. 4 shows two examples of the outputs from our Deep Learning model for person detection and action recognition. The white text on top of each box represents its confidence score for the box location, and the bracketed number next to each action represents the corresponding confidence for that action. The confidence scores may help operators to better judge the result and make better decisions.

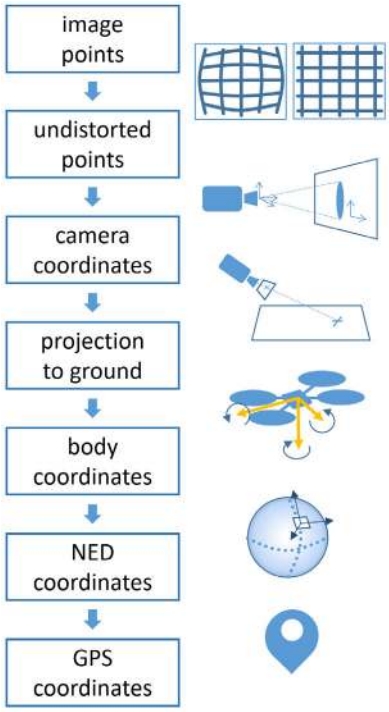


FIGURE 5. Steps of pixel to GPS conversion.

C. PIXEL2GPS CONVERTER

Most UAVs come equipped with GPS receivers, so it is trivial to know where a UAV is positioned when it detects a person. But since the person is typically not directly below the UAV, the GPS position of the person is not the same as the UAV's position. The purpose of the Pixel2GPS converter is to locate detected people by using a spatial conversion of their pixel position in the frame (image) output given by the Deep Learning module to the corresponding GPS position.

Our converter is based on [35], and takes as input the GPS coordinates of the UAV, its height above ground relative to takeoff position, the UAV's roll, pitch and yaw, the camera's pitch angle relative to the UAV, and the image 2D point to be converted. Additionally, it uses the camera's intrinsic matrix and its lens distortion parameters, obtained through calibration with OpenCV [36], which were not considered in [35].

The main steps of the process are shown in Fig. 5. Given a 2D point in the image space, in pixels, we obtain the 3D point by using the camera matrix and distortion parameters. This point is represented relative to the camera referential, and lies in front of the camera sensor. To obtain the corresponding target point on the ground, we consider a line passing through the camera referential origin and this 3D point, and intersect it with the ground plane. Once the target point is obtained in the camera referential, it is then converted to the UAV's NED referential, with the axes pointing at North, East and Down, and the origin in the UAV's center of mass.

Finally, the point is converted to GPS: latitude and longitude in degrees. Unlike [35], we do not use the Earth-centered-Earth-fixed/NED referential conversions, but rather a scaling factor described in [37], which is simpler to implement. The formula for the lengths of one degree of latitude and longitude in meters given in [37] are correct up to 0.01 m. Finally, the North and East components of the NED point are scaled to degrees, and added to the UAV's GPS coordinates.

To test the Pixel2GPS module, we let 3 people stand on a flat baseball field, and took pictures of them using a DJI Phantom 4 UAV, while logging the UAV's telemetry. To the best of our knowledge, the height that DJI's telemetry reports is an integration of the barometer and inertial measurement unit (IMU) values. This makes the height reading more accurate than the GPS altitude reading, which has a very low accuracy of around 30 meters, and would not be adequate for Pixel2GPS. We also corrected the UAV's heading value with the local magnetic declination (the difference between magnetic north and true north), for accuracy. Each person on the ground was carrying a smartphone logging their GPS position. However the accuracy of this method was very low (20 meters), so it could not be used as ground truth. Instead, we got the GPS coordinates from Google Maps, by visually comparing our photos with the satellite image. For this evaluation, the pixel position of each person in each picture was tagged manually, at the person's feet. In total, we captured 103 pictures, with different UAV positions, heights (15 m to 50 m) and camera angles (25 degrees to 90 degrees).

We computed each person's GPS position with our Pixel2GPS converter and compared it with the ground truth GPS. Our error metric was the difference between these two positions, converted to meters. Considering the 103 pictures (309 samples, see Fig. 6), we obtained an average error of 4.5 m, maximum error of 12.8 m, and standard deviation of 3.17. This test demonstrates that our Pixel2GPS module works as expected. We are able to locate people, by using their pixel positions and the UAV's meta data. The method is also very fast: a single conversion takes 0.07 ms on the NVIDIA Jetson TX2, the target hardware.

Currently, the Pixel2GPS module assumes that the ground is flat, which means that the method will not work well in uneven terrain. It also assumes that the target area is at the same altitude as the takeoff site, which may not be true, especially in longer-range missions. The module could be improved in the future with a model of the terrain where the UAV is flying, similarly to [35].

D. USER INTERFACE

For visualization, we developed a web application (see Fig. 7) to display, in real-time, the information broadcast by the onboard application running on the UAV. This web application features a map user interface where the latest drone telemetry and the latest detection details are displayed.

The UAV's telemetry is used mainly to display the UAV position and its camera's field-of-view (where the UAV is 'looking at') on the map. This field-of-view is computed onboard by converting the four pixel corners of the image using Pixel2GPS. Other telemetry information relates to battery level, height above ground relative to the takeoff

FIGURE 6. Results obtained with Pixel2GPS from our test data. Different colors represent different persons. The "G" marker is the ground truth GPS position of each person, placed by visual inspection in Google Maps. The satellite image is by Google.

FIGURE 7. PAL system web application interface.

position, heading of the UAV relative to north and additional information such as the camera angle, is displayed on the bottom panel.

The number of people detected is displayed on the top left of the page, so the operator can have an immediate feedback on how many people are being detected at the moment. Moreover, the people detected are also displayed on the map as blue circle icons representing their position in the world. The recognized actions for each person are displayed as labels above the icons. Additionally, there is the option that allows the operator to individually set the threshold (i.e. confidence level) for displaying primary and secondary actions.

The example shown in Fig. 7 captures a moment where two people are detected, inside the field-of-view projection of the UAV's camera, and their positions are shown. One person is reported with actions of "Standing" and "Reading" with a confidence level of 80% and 55%, respectively. The other person is reported with the action "Walking" with a confidence level of 96%.

IV. FIELD TEST OF THE PAL SYSTEM

We conducted several field tests as a proof of concept that our Person-Action-Locator system can indeed perform multiple human detection, recognize their primary and secondary actions and calculate their real-world positions in near real-time onboard of the UAV. All field tests were conducted in daylight, on a baseball field of Okutama town, located west of Tokyo, Japan. From the original test data [2], we selected

- 3 primary actions: Walking, Standing, Sitting;
- 3 secondary actions: Reading, Carrying, Pushing / Pulling.

A. PAL SYSTEM PROTOTYPE

The main purpose of this field test was to validate our system integration and interoperability. Therefore, we focused on a simple scenario, that allowed us to observe in near real-time the consistency between the information displayed on the PAL Web Application (originally reported by the onboard PAL system) and the real scenario. Additionally, we collected data such as the field of view of the camera, the number of people detected and their calculated position, as well their recognized actions. System specifications, such as the size of data sent to the server, the time of overall refresh rate of the system, and battery life, were also measured.

We will start by explaining our prototype hardware configuration, before we explain the test scenario.

1) PROTOTYPE HARDWARE CONFIGURATION

The PAL onboard application is deployed on the NVIDIA Jetson TX2 (8 GB RAM), that we use as the onboard computer for the DJI Matrice 100 (see Fig. 8). The TX2, which is mounted on the Auvidea J120 carrier board, connects to the UAV using a USB to TTL/UART connection. The UAV is powered by one DJI TB48D battery, of 130 W·h (W·h = watt-hour = 3600 J), that we also use as a power source to our onboard computer via a custom-made voltage regulator.

The PAL onboard application supports the DJI Matrice 100 by integrating the DJI Onboard SDK. For the camera, we use a Logitech C920 USB webcam, which is attached to the UAV with a pre-defined fixed angle. Internet connectivity is achieved with a PIX-MT100 LTE dongle, which allows our onboard system to output the onboard results to a server deployed on Amazon Web Services (AWS) for real-time visualization displayed on the PAL Web Application.

The total payload of our onboard setup used to run our PAL system (including onboard computer, webcam, power adapter and cables) is 0.57 kg.

An obvious limitation of this hardware configuration is the camera setup. This limitation arises from the fact that the DJI Matrice 100 and all its compatible DJI gimbal-equipped cameras do not support a live feed API that we could use on our onboard application. Therefore, we use a regular USB camera without gimbal, that allows us to open a live feed through OpenCV, in order to support the desired near real-time behavior of the PAL system.

2) SCENARIO SETUP AND OBSERVATIONS

Two actors performed the selected primary and secondary actions. 500 frames were retrieved corresponding to approximately 12 minutes of scenario observation. Due to the camera

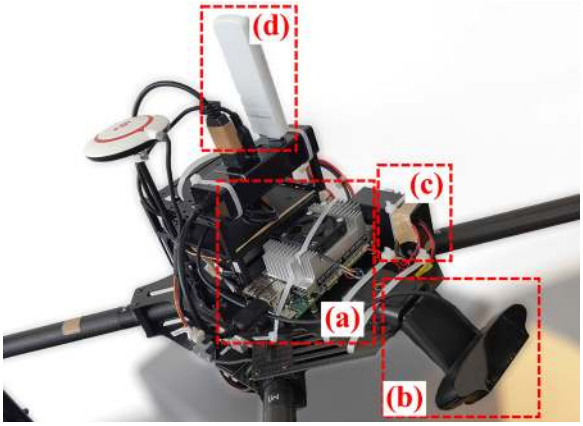


FIGURE 8. Matrice 100 drone used for the PAL system prototype. (a) NVIDIA Jetson TX2, (b) Logitech C920 webcam, (c) voltage regulator, (d) LTE dongle.

limitation explained above, we kept the UAV hovering at all times at a fixed position, around 32 meters above ground relative to takeoff position.

We observed a 3 seconds latency of the information in the web interface. The dominant factor is the onboard deep learning inference time (1.3–2.2 s), while the latency between the UAV and the Web Application via our AWS server was 30–60 ms, and the Pixel2GPS has negligible computation time (< 1 ms). After each inference, the UAV's onboard computer transmits a message of 230–500 bytes, depending on how many people were detected. This is significantly smaller than sending the camera image, which can be 200 kB or more, and thus too much if bandwidth is limited.

We observed the battery life of the system over multiple flights. On average, the TX2 was measured to consume 4.2 W of power running the PAL system, which translates to just 0.98 W·h. The average flight time with our described prototype setup (0.57 kg payload) was about 14 minutes. For this specific drone and this specific battery, DJI officially reports a performance of up to 20 minutes of hovering time with a 0.5 kg payload and up to 16 minutes of hovering time with a 1.0 kg payload.

From this we conclude that the energy consumption of our onboard system is minimal, and hence existing UAV platforms meet the energy requirements of the whole system.

Person detection on the flat baseball field was near perfect (mAP of 99%), under perfect conditions, i.e., the only visible objects on the field were the actors. Regarding action recognition, we observe mAP of 48.6% for primary actions and 29.6% for secondary actions. Here the main insight is that action recognition worked, while the relative values deserve interpretation due to the highly artificial setup.

B. PAL SYSTEM SURVEILLANCE FIELD TEST

The main purpose of this field test was to evaluate the person detection and action recognition of our PAL system, without the limitations mentioned on our prototype configuration in Sect. IV-A. Therefore, we used the DJI Phantom 4 and its built-in gimbal equipped camera to scan a target area (here, a baseball field) in a crisscross pattern. For this test, we also designed a test scenario that better represents real use-cases, such as search and rescue or surveillance.

1) SURVEILLANCE TEST SCENARIO

Three actors performed a subset of the primary and secondary actions (see Sect. IV). Each act lasted for around 150 seconds. After Act2 finished, they repeated the two Acts, for a total length of 10 minutes. PA abbreviates "primary action" and SA abbreviates "secondary action".

Act1	Actor 1 sits (PA) and reads (SA)
	Actor 2 walks (PA) while pushing a wheelbarrow (SA)
	Actor 3 stands (PA)
Act2	Actor 1 sits (PA) and reads (SA)
	Actor 2 stands (PA)
	Actor 3 walks (PA)
Act1	(repetition)
Act2	(repetition)

During the entire test, when the UAV pilot is visible, he considered as standing (PA) and carrying (SA), as he is holding the UAV controller.

Data was obtained as follows:

- Perform the scenario one time;
- Retrieve 10 minutes of video at 30 FPS using the Phantom 4;
- Extract one frame every 1.5 seconds and obtain 327 frames, ignoring takeoff and landing.

2) RESULTS

The results are shown in Table 2. For human detection, we obtained a mAP of 60.9% in the surveillance test, compared to 85.0% in the benchmark (see Sect. III-B.2). The decrease in performance due to false positives arises when the UAV's camera is pointing outside the baseball field (see Fig. 9). Whereas in our training data, the videos are split into chunks of camera footage that focus only on its actors, in the case of our field test we consider an entire flying sequence. Therefore, our field test's footage comprises of the raw uninterrupted video, possibly with fluctuating footage that is not present in the training data.

TABLE 2. Results of person detection and action recognition, considering only selected 3 primary and 3 secondary actions (see Sect. IV). "SFT" abbreviates "Surveillance field test".

	Detection			Action (mAP)	
	Recall	mAP	F1	PA (3)	SA (3)
Benchmark	86.0	85.0	84.0	38.8	44.6
SFT	83.0	60.9	67.6	26.5	17.9

The accuracy of action recognition depends on the accuracy of the person detection, since actions can only be recognized if the person was detected. This situation has an negative effect on the mAP of action recognition.

FIGURE 9. Example of 4 false positives, in red, detected along a fence in the surveillance field test. The fifth detection, in green, is a true positive.

V. CONCLUSION

In this paper, we present a UAV-based situational awareness system, called Person-Action-Locator (PAL), which can be useful for tasks such as search and rescue or surveillance. The main components of the PAL system are (1) the Deep Learning component that automatically detects people and recognizes their actions, (2) the Pixel2GPS converter that estimates the GPS position of persons by image processing, and (3) the PAL interface that visualizes detected people and actions on a map. The integration of all components was successfully tested in the field. The Deep Learning models have also been rigorously evaluated in our laboratory.

An important aspect of our Deep Learning model for person detection (POINet) is that our model does not rely on assumptions of the orientation or the aspect ratio of the bounding boxes. The Deep Learning model for action recognition demonstrated improved results due to the consideration of previous frames.

We expected that computing hardware on the UAV would be a main limitation for the Deep Learning model, since the frame rate on the NVIDIA Jetson Tegra X2 (0.667 FPS) is lower than the frame rate used to build the model (3 FPS). However, our results suggest that the effect is small.

As a system for situational awareness, a visual camera is limited to detecting objects that can actually be seen. In the future, we might want to consider other sensors, such as a thermal camera. The Pixel2GPS module can also be improved with a digital elevation model, in order to work even when the area of interest is at a different altitude than the takeoff area, or in target areas that are not flat.

In conclusion, we have demonstrated that the Person-Action-Locator (PAL) system is a practical solution for situational awareness tasks that involve the detection and location of people, and recognition of their actions. We achieved strong results with limited computing performance onboard the UAV. Thus our solution does not depend on powerful communication infrastructure, which cannot be taken as granted in disaster or surveillance situations. We will continue to improve our solution with better hardware.

REFERENCES

[1] S. D. Ramchurn, T. D. Huynh, F. Wu, Y. Ikuno, J. Flann, L. Moreau, J. E. Fischer, W. Jiang, T. Rodden, E. Simpson, S. Reece, S. Roberts, and N. R. Jennings, "A disaster response system based on human-agent collectives," *J. Artif. Intell. Res.*, vol. 57, pp. 661–708, Dec. 2016.

[2] M. Barekatain and M. Marti, H.-F. Shih, S. Murray, K. Nakayama, Y. Matsuo, and H. Prendinger, "Okutama-action: An aerial view video dataset for concurrent human action detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 2153–2160.

[3] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors*, vol. 37, pp. 32–64, Sep. 1995.

[4] K. Phillips, M. J. Longden, B. Vandergraff, W. R. Smith, D. C. Weber, S. E. McIntosh, and A. R. Wheeler, "Wilderness search strategy and tactics," *Wilderness Environ. Med.*, vol. 25, no. 2, pp. 166–176, Jun. 2014.

[5] M. B. Bejiga, A. Zeggada, A. Nouffidj, and F. Melgani, "A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery," *Remote Sens.*, vol. 9, no. 2, p. 100, 2017.

[6] C. Sampedro, A. Rodriguez-Ramos, H. Bavle, A. Carrio, P. de la Puente, and P. Campoy, "A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques," *J. Intell. Robotic Syst.*, vol. 95, no. 2, pp. 601–627, Aug. 2019.

[7] G.-R. Shih, P.-H. Tsai, and C.-L. Lin, "A speed up approach for search and rescue," in *Proc. IEEE Int. Conf. Syst., Man, (SMC)*, Oct. 2018, pp. 4178–4183.

[8] R. T. Collins, A. J. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," *Proc. IEEE*, vol. 89, no. 10, pp. 1456–1477, Oct. 2001.

[9] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, and S. Pankanti, "Smart video surveillance: Exploring the concept of multiscale spatiotemporal tracking," *IEEE Signal Process. Mag.*, vol. 22, no. 2, pp. 38–51, Mar. 2005.

[10] R. Borja, J. de la Pinta, A. Álvarez, and J. M. Maestre, "Integration of service robots in the smart home by means of UPnP: A surveillance robot case study," *Robot. Auton. Syst.*, vol. 61, no. 2, pp. 153–160, Feb. 2013.

[11] J. López, D. Pérez, E. Paz, and A. Santana, "Watchbot: A building maintenance and surveillance system based on autonomous robots," *Robot. Auton. Syst.*, vol. 61, no. 12, pp. 1559–1571, Dec. 2013.

[12] L. Geng, Y. F. Zhang, P. F. Wang, J. J. Wang, J. Y. H. Fuh, and S. H. Teo, "UAV surveillance mission planning with gimbaled sensors," in *Proc. 11th IEEE Int. Conf. Control Autom. (ICCA)*, Jun. 2014, pp. 320–325.

[13] C. Hong and D. Shi, "A control system architecture with cloud platform for multi-uav surveillance," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov.*, Oct. 2018, pp. 1095–1097.

[14] E. Semsch, M. Jakob, D. Pavlicek, and M. Pechoucek, "Autonomous UAV surveillance in complex urban environments," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. Intell. Agent Technol.*, vol. 2, Sep. 2009, pp. 82–85.

[15] E. Semsch, M. Jakob, and D. Pavlí ek, and M. P chou ek, "Occlusion-aware multi-UAV surveillance," in *Proc. 9th Int. Conf. Auto. Agents Multiagent Syst. (AAMAS)*, vol. 1. Richland, Washington, USA, 2010, pp. 1407–1408.

[16] Z. Beck, W. L. Teacy, A. Rogers, and N. R. Jennings, "Collaborative online planning for automated victim search in disaster response," *Robot. Auton. Syst.*, vol. 100, pp. 251–266, Feb. 2018.

[17] M. Aljehani and M. Inoue, "Performance evaluation of multi-UAV system in post-disaster application: Validated by HITL simulator," *IEEE Access*, vol. 7, pp. 64386–64400, 2019.

[18] W. Tao, Y. Lei, and P. Mooney, "Dense point cloud extraction from UAV captured images in forest area," in *Proc. IEEE Int. Conf. Spatial Data Mining Geograph. Knowl. Services*, Jun. 2011, pp. 389–392.

[19] T. Arnold, M. De Biasio, A. Fritz, and R. Leitner, "UAV-based measurement of vegetation indices for environmental monitoring," in *Proc. 7th Int. Conf. Sens. Technol. (ICST)*, Dec. 2013, pp. 704–707.

[20] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, vol. 1, no. 1, pp. 886–893.

[21] O. Tuzel, F. Porikli, and P. Meer, "Human detection via classification on Riemannian manifolds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[22] L. Xia, C.-C. Chen, and J. K. Aggarwal, "Human detection using depth information by Kinect," in *Proc. CVPR Workshops*, Jun. 2011, pp. 15–22.

[23] W. Ouyang and X. Wang, "Joint deep learning for pedestrian detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2056–2063.
[24] Y. Tian, P. Luo, X. Wang, and X. Tang, "Deep learning strong parts for pedestrian detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1904–1912.
[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.
[26] V. Ramanathan, J. Huang, S. Abuelhaija, A. Gorban, K. Murphy, and L. Fei-Fei, "Detecting events and key actors in multi-person videos," in *Proc. IEEE CVPR*, Jun. 2016, pp. 3043–3053.
[27] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.
[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[30] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 21–37.
[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[33] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, and A. Courville, "Recurrent batch normalization," 2016, *arXiv:1603.09025*. [Online]. Available: https://arxiv.org/abs/1603.09025
[34] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2016, pp. 1933–1941.
[35] M. G. Johnston, "Ground object geo-location using UAV video camera," in *Proc. IEEE/AIAA 25th Digit. Avionics Syst. Conf.*, Oct. 2006, pp. 1–7.
[36] G. Bradski, "The OpenCV library," *Dr. Dobb's J. Softw. Tools*, vol. 25, no. 11, pp. 120–126, 2000.
[37] R. H. Rapp, *Geometric Geodesy Part I*. Columbus, OH, USA: Ohio State Univ. Department of Geodetic Science and Surveying, 1991. [Online]. Available: https://kb.osu.edu/handle/1811/24333

RÚBEN GERALDES received the B.Sc. degree in information system and computer engineering from the Instituto Superior Tecnico, Universidade Lisboa, Lisbon, Portugal. He was a JST Project Researcher on a big data project with the National Institute of Informatics. He is currently a Specialist Researcher with the National Institute of Informatics, Tokyo, Japan, working on the research and development of a UTM system for the low-altitude airspace in Japan. His current research interests include computer graphics and multimedia systems, distributed systems and IT systems, and human-agent interactions.

ARTUR GONÇALVES received the master's degree in electrical and computer engineering from the Instituto Superior Técnico, University of Lisbon, Portugal, in 2015. Since 2016, he has been with the National Institute of Informatics, Tokyo, Japan, where he integrates a team working on unmanned aircraft systems traffic management (UTM) research and system development. He focused on developing system prototypes and visualization tools. His research interests include computational geometry and cyber-physical systems.

TIN LAI received the B.Sc. degree in computer science and B.C.E. degree (Hons.) from The University of Sydney, Australia, in 2017, where he is currently pursuing the Ph.D. degree with the Machine Learning and Robotics Lab, School of Computer Science.

His current research interests include statistical machine learning techniques, multiagent systems, motion planning, and probabilistic predictions with applications in robotics.

MATHIAS VILLERABEL received the B.S. degree in computer science and mathematics from Sorbonne University, France, in 2016. He is currently the master's degree in computer science from Sorbonne University, France. His studies are centered around artificial intelligence and multi-agent systems and their use in solving complex problems.

WENLONG DENG received the B.Sc. degree (Hons.) in electrical engineering from the University of Electronic Science and Technology of China, in 2017. He is currently pursuing the master's degree with the School of Electronic Engineering, EPFL, Switzerland. His current research interests include machine learning and computer vision for the human-centered application.

ANA SALTA received the B.S. and M.Sc. degrees in information systems and computer engineering from the Instituto Superior Tecnico, University of Lisbon, Portugal, in 2015 and 2017, respectively. Her current research interests include intelligent systems, artificial agents and multiagent systems, and games.

KOTARO NAKAYAMA received the Ph.D. degree in information science and technology from Osaka University, Osaka, Japan, in 2008. He has careers in both academic and industrial area including; a CEO of the Kansai Informatics Institute; an Assistant Professor of CSK and The University of Tokyo; and an Assistant Professor with the Graduate School of Engineering, The University of Tokyo. He is currently a CEO of NABLAS, Inc., and a member of the Matsuo Lab, Graduate School of Engineering, The University of Tokyo. He published nine computer science related books including *Data Science Bootcamp* and translation of Y. Bengio's Deep Learning.

YUTAKA MATSUO received the B.S., M.S., and Ph.D. degrees from The University of Tokyo, in 1997, 1999, and 2002, respectively. After working at the National Institute of Advanced Industrial Science and Technology (AIST) and Stanford University, he joined the Faculty of The University of Tokyo, in 2007. He is currently a Professor with the Graduate School of Engineering, The University of Tokyo. From 2012 to 2014, he served as an Editor-in-Chief and, from 2014 to 2018, as the Chair of the ELSI Committee at the Japan Society for Artificial Intelligence (JSAI). He is currently the President of the Japanese Deep Learning Association (JDLA).

HELMUT PRENDINGER received the master's and Ph.D. degrees in logic and artificial intelligence from the University of Salzburg, Austria. He held positions as a Research Associate and as a JSPS Postdoctoral Fellow with The University of Tokyo. In 1996, he was a Junior Specialist with the University of California at Irvine, Irvine. He is currently a Full Professor with the National Institute of Informatics, Tokyo. His team contributes to developing the entire UTM system as part of a large-scale Japanese government project. He has published more than 230 refereed papers in international journals and conferences. His h-index is 38. His current research interests include unmanned aircraft systems traffic management (UTM) and machine learning (ML), especially deep learning for drone use cases.