



Autonomous long-range drone detection system for critical infrastructure safety

Xindi Zhang¹ · Kusrini Kusrini²

Received: 6 March 2020 / Revised: 8 August 2020 / Accepted: 9 December 2020 /
Published online: 12 February 2021
© The Author(s) 2021

Abstract

The development of unmanned aerial vehicles has been identified as a potential source of a weapon for causing operational disruptions against critical infrastructures. To mitigate and neutralise the threat posed by the misuse of drones against malicious and terrorist activity, this paper presents a holistic design of a long-range autonomous drone detection platform. The novelty of the proposed system lies in the confluence between the design of hardware and software components to effective and efficient localisation of the intruder objects. The research presented in the paper proposes the design and validation of a situation awareness component which is interfaced with the hardware component for controlling the focal length of the camera. The continuous stream of media data obtained from the region of vulnerability is processed using the object detection that is built on region based fully connected neural network. The novelty of the proposed system relies on the processing of multi-threaded dual-media input streams that are evaluated to mitigate the latency of the system. Upon the successful detection of malicious drones, the system logs the occurrence of intruders that consists of both event description and the associated media evidence for the deployment of the mitigation strategy. The analytics platform that controls the signalling of the low-cost sensing equipment contains the NVIDIA GeForce GTX 1080 for detecting drones. The experimental testbeds developed for the validation of the proposed system has been constructed to include environments and situations that are commonly faced by critical infrastructure operators such as the area of protection, drone flight path, tradeoff between the angle of coverage against the distance of coverage. The validation of the proposed system has resulted in yielding a range of intruder drone detection by 250m with an accuracy of 95.5%.

Keywords Drone detection · Deep-learning · Infrastructure security · Sensing equipment

✉ Xindi Zhang
xindi.zhang@qmul.ac.uk

1 Introduction

Unmanned aerial vehicles (UAV), also known as drones, have developed rapidly in recent years. Nowadays, companies such as Amazon, Alibaba, and even pizza chains are pushing forward to use drones, for service provision such as package and food delivery. However, the high-speed mobility and behavior dynamics of UAVs need to be monitored to detect and subsequently, to deal with rogue drones piloted with malicious intent. The misuse of drones can be a huge threat not only to the safety of property but also human lives. Because of the increasing number of drones terrorism, malicious and illicit activities, it is necessary to detect the drone before it getting close to people or buildings [2].

To address the challenge of detecting small objects taken from a distance, the use of deep learning models has been reported in the literature, such as AZNet [20], TridentNet [16], SNIPER [25]. The computational complexity of these methods limits their integration into a UAV detection platform in a real-world operating environment. However, in the real-world context of the installation of such sensors to secure critical infrastructures, real-time detection of objects on the horizon to allow the deployment of countermeasures to mitigate threats is critical. A framework of drone detection and tracking has been proposed in [31], which can run in real-time. However, for small drones at great distances, these deep learning-based object detection networks are difficult to detect. Therefore, this paper proposes a dual camera system that combines traditional computer vision algorithms and deep learning algorithms to achieve real-time long-range UAV detection and tracking.

An example of the region to be protected against intruder drone is presented in Fig. 1. The security region is categorised into three namely (i) safe zone; (ii) alert zone and (iii) mitigate zone. The safe zone corresponds to the perimeter region, in which the presence of any object does not present a danger to the operation of the critical zone. The region between the safe and alert zone represents an increase in the threat level, and any object penetrating the perimeter of the safe zone should be alerted of its presence to the infrastructure command center. Subsequently, the violation of the alert zone is considered to pose an imminent threat to the infrastructure and thus will mitigation action should be executed to neutralise the threat before the perimeter of the critical region is penetrated.

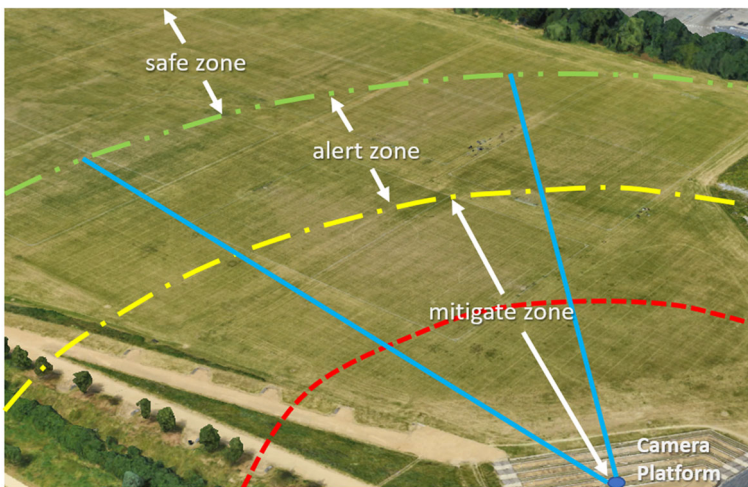


Fig. 1 The protection area of the camera system

Addressing the need to develop a system capable of being autonomously operated for detecting the presence of malicious objects in the vicinity of critical infrastructure, the research in this paper presents a holistic system that consists of both software and hardware components. The software component consists of situation awareness module for identifying intruder objects appearing on the horizon and the object detection framework based on the Region based Fully Connected Neural Network (RFCN) that can detect the drone. The hardware component is a low-cost sensor consisting of a pan-tilt-zoom (PTZ) platform controlled by servo motors and equipped with a camera capable of being programmatically controlled for focal length changes. The signalling of the hardware equipment is triggered by the software component based on a state machine that transitions between the scanning mode and the detection modes of operation. The state machines are designed to handle the high-speed manoeuvrability of the drones and being piloted across a three-dimensional coordinate system.

The research contributions of the paper are:

- design and implementation of a situational awareness module that can monitor the horizon for intruder appearance;
- to develop a hardware signalling module that temporally synchronises the situation awareness module and the object detection component;
- to identify the intruder object based on a multi-class deep-learning network using RFCN and
- to estimate the distance of the intruder object and alert the command center for implementing countermeasures to neutralise the threat of intruder objects.

The rest of the paper is structured as follows. In Section 2, a detailed review of various drone detection systems reported in the literature is presented. The review includes the analysis of long-range radar system for completeness. However, as the scope of the research presented in the paper relates to the application of computer vision technologies, a detailed analysis of the shortcomings from the techniques reported in the literature are further analysed. Subsequently, in Section 3, an overview of the proposed dual-camera system for autonomous long-range drone detection is presented. The section outlines the various software components that are designed to interface with the underlying hardware equipment. Section 4 presents in detail the performance evaluation of the proposed system carried out in support of critical infrastructure operators. The discussion on the distance estimate approaches is considered in Section 5. The conclusion and future work are presented in Section 6.

2 Literature review

The remotely piloted autonomous systems (RPAS) and the UAVs undertake unpredictable computer-controlled movements, that varies in speed and exploit the manoeuvring capabilities of the physical object. Their resemblance to other aerial objects such as birds, aeroplanes, result in challenges attributed to automatic detection, identification and most importantly accurate localisation in the 3D space. In order to solve this problem, several types of sensors (RF, GPS, radio waves, radar, acoustic etc.) have been proposed in the literature and subsequently been deployed for drone detection [28]. While the deployed sensors can identify all the objects, it is important to recognise the specific malicious drone that might be approaching the Critical Infrastructures (CI) with the intention of creating harm.

In this regard, the application of computer vision techniques has resulted in improved performance as reported in [10]. This is primarily attributed to the recent growth in the deep learning technologies that have been shown to be very powerful in computer vision tasks. In the rest of the section, an outline of various approaches proposed in the literature is summarised.

Traditionally, the object detection task has been to classify regions of any predefined object in a training dataset. Early drone detection attempts used a similar approach to detect whether or not the image region consisted of drones. In this case, a computer vision approach was used to select an appropriate representation of the object using handcrafted features.

The most successful approaches using handcrafted features require bag of visual words (BoVW) was reported in [26] that includes representations of the objects with the help of local feature descriptors such as scale invariant feature transform (SIFT) [19], speeded-up robust features (SURF) [1], and histogram of oriented gradients (HOG) [6]. After training a discriminative machine learning (ML) model, e.g., support vector machines (SVM) [4], with such representations the images are scanned for the occurrence of learning objects with sliding window technique. These reported methods have two crucial drawbacks. The first one is that the features have to be crafted well for the problem domain to highlight and describe the important information in the image. The handcraft features are not robust enough for complex situations such as illumination variations, deformations, scale variations and so on. The second one is the computational burden of the exhaustive search done by the sliding window technique. The burden of sliding window waste too much time on background region which cannot run in real-time.

With the extremely large progress made by deep learning algorithms in the field of image classification tasks, similar methods have begun to be used to address target detection problems. These techniques can be divided into two categories: methods based on a region proposal and methods based on a single shot.. The first class of methods differs from traditional methods in that it uses convolutional neural networks (CNNs) to learn the features of regions extracted through selective search or region proposal networks, as reported in [8, 9, 23]. In the single shot approach, the goal is to directly compute the bounding boxes of objects in the image, rather than processing the regions of the image. One approach uses CNNs to extract multiscale features and combine them to predict the bounding box as proposed in [13, 18]. Another approach, called You Only Look Once (YOLO), reported in the literature [22], divides the final feature map into a two-dimensional grid and uses each grid cell to predict a bounding box. The techniques mentioned originate from the overall goal of object detection and are not explicitly qualified to tackle the drone classification.. A large number of datasets that are trained on well-known objects that are frequently encountered in real-life are used to develop a framework for detection.

While the previously reported approaches considered the use of feature extracted from a single frame appearance of drones, the use of motion and spatio-temporal features has also been studied. In particular, the approach that first creates spatio-temporal cubes using sliding window method at different scales, applies motion compensation to stabilise ST cubes, and finally utilises boosted tree and CNN based regressors for bounding box detection [24]. In addition, Multi-frame representation of a drone is learnt by convolutional long short-term memory (ConvLSTM) in [30]. The cross-correlation layer helps to generate search windows and localize the drone.

Earlier approaches reported in the literature, have focused on detection accuracy with existing datasets. However, those approaches have failed to specify the detection range of

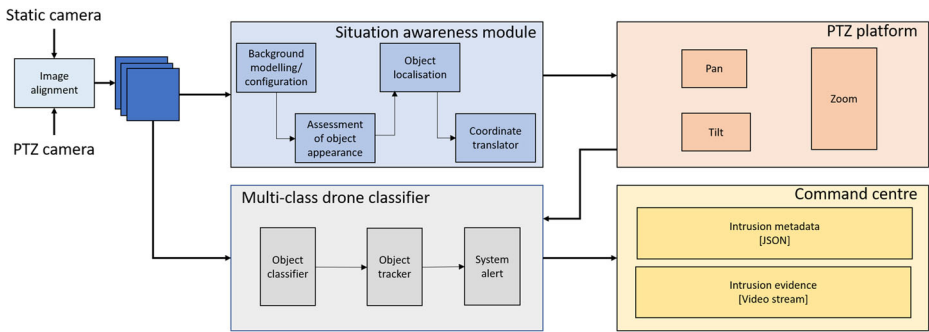


Fig. 2 Proposed dual camera system

distance and the operation speed which are important in practical applications. Vision-based approaches depend on the appearance feature of the interest target. For long-range target, the pixels shown in the video are too less to offer enough features. The target is too small to be detected by the algorithm. In this paper, a real-time holistic system is demonstrated to detect intruder drone appear in the horizon from a far distance against a security area with the help of dual-camera corporation.

3 Proposed system

As presented in Fig. 1, the functional specification of the system requires the installation of the drone detector at the perimeter of the critical infrastructure able to protect against imminent attacks exploiting the vulnerable pathways, through which the malicious drone could be piloted. Addressing the real-world needs of the operational environment, the conceptual design of the proposed solution is presented in Fig. 2.

The system implementation contains four components, namely (i) situation awareness module, (ii) PTZ platform, (iii) multi-class drone classifier using deep-learning, and (iv) alert command center. In the beginning, there are two cameras, a static camera and a PTZ camera, streaming their video to the system and being aligned at first. The situation awareness module is operating based on the static camera videos to monitor intruders. The PTZ platform consists of a pan-tilt platform with the camera that supports programmatic control of the focal length of the lens.

The Raspberry Pi processes the pan-tilt-zoom (PTZ) signals to enable the appropriate binary bits that are interfaced with the servo motors to position the platform precisely. The continuous media stream captured by the camera is transmitted to the analytical component, where the media is processed using the deep-learning network framework for drone detection, and the intruders on the horizon are then identified. When the intruder is identified as a drone, the identification situation and intrusion proof will be reported by the command centre. Figure 3 shows the various operating states involved in the drone detection platform. The installation phase involves the installation of the detector at infrastructure perimeter. The next step is to configure the operating environment of the detector, which includes aligning two cameras to identify the region of attack against the background. Subsequently, the analytics component was considered to be two operating states and a third state for signalling

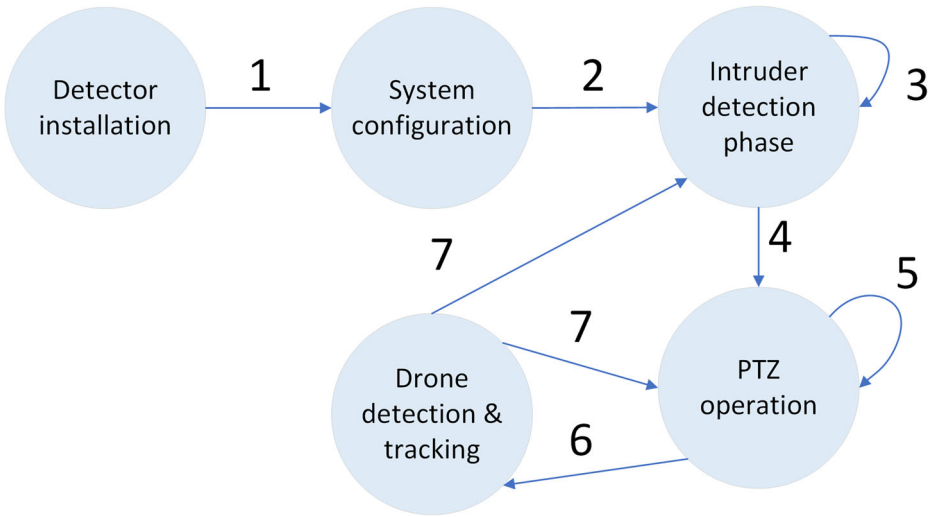


Fig. 3 Operational states of the proposed system

Raspberry Pi with instructions for PTZ. The rest of the section provides a detailed outline of the various state transitions that were implemented for the proposed system to operate.

3.1 System configuration for the region of protection

The equipment configuration is a critical step in ensuring the successful operation of the proposed system against a malicious drone attack. It involves the modelling of foreground and ground objects against which the malicious drone could not be piloted. Such foreground objects include trees, grass and any man-made structure (such as energy distribution poles). The rationale for such a premise is based on the fact critical infrastructures are operated away from the urban population due to safety regulations in farmlands. Therefore, the number of intruder objects that could create harm against infrastructures are limited. In most cases, the appearance of birds, aeroplanes, helicopters and people are a common sight. Therefore, to model the background the captured video stream is structured into a 20x20

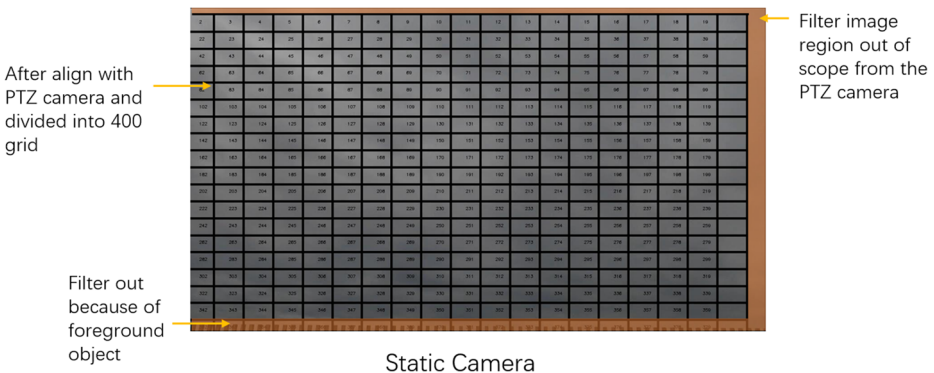


Fig. 4 Detector configuration of the background objects

grid as presented in Fig. 4. The grid offers the security personnel to identify the regions containing the foreground and natural objects, through which the drone could not be piloted. The marked regions are ignored through the use of the binary mask. The binary mask is a black and white image, the grids selected by security personnel are drawn in black, the remain regions are drawn in white. After multiplying the binary mask with the frame, any movement in the foreground regions will be ignored. The unfiltered region that presents the vulnerability through the point of attack from the drone is further processed by the analytics component. For the dual-camera system, the intruder detection and zoom into the region are operated by two different cameras. To use the static camera to guide the PTZ operation to reach a certain grid, the scene they captured should be aligned at the beginning. The method used is Brute-force matcher which match feature point of two different images and localize the matched region. Based on the matched region, the area is divided into 400 grid for further detection and guidance. An example is shown in Fig. 4, the view of the static camera is aligned with the PTZ camera and divided into 400 grid. The area doesn't align and the grids contain foreground object are filtered out by the binary mask.

3.2 Situation awareness module

Following the configuration, the static camera enters the intruder detection phase, in which the detector continues to scan the horizon for any potential intruder objects. As the objective of the detector is to provide security for the maximum field of view, the focal length of the camera is maintained at a minimum. Therefore, the amount of pixel variations captured by the camera is limited to a small region in the image, with effectively few pixel changes. With such a level of low quantity of information available, the deep-learning networks are not able to successfully process the video resulting in the detection. Therefore, the proposed system uses one of the classical vision-based techniques for edge detection namely Canny algorithm to identify the potential objects appearing on the horizon. The operation of the algorithm is summarised in (1), which uses a Gaussian filter to smooth the image and remove noise.

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i - (k + 1))^2 + (j - (k + 1))^2}{2\sigma^2}\right); \quad (1)$$

$$1 \leq i, j \leq (2k + 1). \quad (2)$$

The equation corresponds to a $(2k + 1) \times (2k + 1)$ Gaussian filter kernel. The kernel size used in this paper is 5×5 with $\sigma = 1$. The kernel is convoluted with the image to remove high-frequency variations. Subsequently, the image is processed with a Sobel kernel across both vertical and horizontal directions. The derivative in vertical direction (G_y) and horizontal direction (G_x) results in the determination of the edge gradient for each pixels:

$$EdgeGradient(G) = \sqrt{G_x^2 + G_y^2} \quad (3)$$

$$Angle(\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \quad (4)$$

Following the assignment of intensity gradient at each pixel in the video sequence, a threshold is applied to distinguish pixels that appear to be a part of the edge. The value of the threshold is empirically selected to preserve the connectivity of continuous pixel sequences against noise introduced by the spatial objects such as clouds. The pixels gradient values below the threshold are ignored as noise and are not considered in subsequent processing of the image sequences.

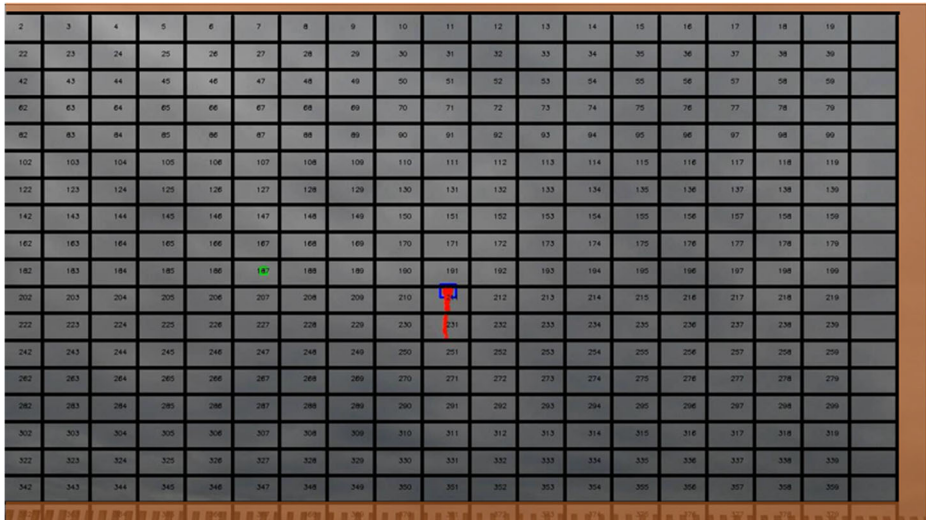


Fig. 5 Intruder object localisation using the edge detection and dilation algorithm

Further to the computation of the edge gradient, the dilation is operated by (5). The maximizing operation enlarges the edges to be an entire foreground. The dilation kernel used is 3×3 .

$$dst(x, y) = \max_{(x',y'):element(x',y') \neq 0} src(x + x', y + y') \tag{5}$$

The bounding box is generated by finding contour of the foreground. The outcome is presented in Fig. 5. In this example, there are two intruders detected in the view of the static camera. The bigger size of the bounding box means the intruder is closer to the infrastructure which has priority to identify. So the biggest bounding box is drawn in blue with a red trajectory. The red trajectory is the connection of the center points of previous bounding boxes. The thickness of the trajectory decays with the increasing number of frames. Other objects are drawn in a green box waiting for the next round of operation.

The bounding box of the intruder object to a specific grid in the image captured with the fixed focal length of the static camera facilitates the analytics component to single the Raspberry Pi and the camera to gather additional information from the respective grid through the pan-tilt and zoom operation.

3.3 PTZ platform signalling and control

The objective of the PTZ operation is to focus on the grid where the intruder has been detected from the static camera and gather additional visual information to be processed by the deep-learning algorithm. To achieve this objective, the following premise has been adopted. The video stream resolution supported by the camera is full HD resolution (1980x1080 at 30FPS) and despite the changes in the focal length, the output of the camera remains the same. The variation of the focal length changes to the Field of View (FoV) is presented in Fig. 6. As the camera is pointed towards the intruder object by progressing increasing the focal length, the FoV narrows that limits the ability to scan large-regions.

Considering the neutral state of the camera position is at 90 degrees for both pan and tilt, a look-up table has been created with pre-encoded values that correspond to the estimated

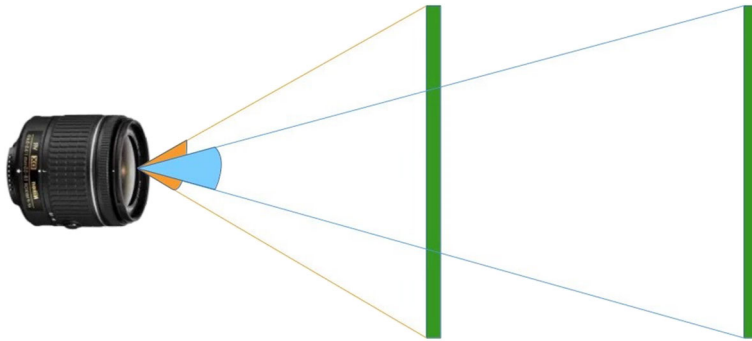


Fig. 6 The impact of focal length variation upon the Camera FoV

changes to be signalled for each of the grid positions as configured in Fig. 4. To reach the certain block, the platform should pan and tilt in relative degree according to specific values (59, 62, 65, 68, 70, 73, 77, 79, 82, 86, 90, 94, 98, 101, 104, 107, 111, 114, 117, 120) for pan and (108, 106, 104, 102, 101, 100, 99, 97, 95, 92, 90, 88, 87, 85, 83, 82, 79, 77, 75, 73) towards tilt. In the operational environment, pan to 65 degrees and tilt to 106 degrees will reach the block 23, pan to 111 degrees and tilt to 77 degrees will reach the block 337.

Besides, the ratio of the intruder object bounding box to the overall video frame has been considered in determining the level of zoom to be used for acquiring addition visual information based on which the drone detection can be operated. The relationship between the size of the intruder object and the correspondent signalling of the focal length variations is presented in Table 1. The ratio is computed between the width of the video sequence and the width of the bounding box marked at the periphery of the intruder object. The signalling parameter to control the focal length of the camera is determined based on the object ratio as identified by the intrusion detection from the static camera.

The focal length variation will influence the FoV of the camera as shown in Fig. 6 and Table 2. Following the focal length changes, the video stream obtained is further processed by drone detection (as presented in Section 3.4). It is also important to note that, the signalling received from the analytics component is processed by the Raspberry Pi, and further triggers the servo motor actions, it is important to note that any change to the camera position results in an unfocused image with video stream containing blurry vision until the camera hardware is re-adjusted using the in build auto-focus functionality. Therefore, the PTZ operation inherently includes a time delay before transitioning for the drone detection

Table 1 Mapping of intruder detection size to the zoom level

size ratio	zoom level
> 200	15
> 150	14
> 100	10
> 75	9
> 50	8
> 10	7
> 5	4
<= 5	0

Table 2 The mapping of FoV against camera focal length

Signalling zoom level	FoV	Camera focal length (mm)
0	54.94	5.27
1	46.53	6.42
2	41.89	7.24
3	36.91	8.34
4	32.21	9.67
5	27.44	11.46
6	22.92	13.83
7	18.05	17.68
8	14.09	22.74
9	10.81	29.73
10	7.66	42.03
11	5.39	59.79
12	4.53	71.16
13	3.78	85.30
14	3.41	94.57

phase of operation. During the time of zoom in and re-adjust, the target may change its position which will be out of view for the PTZ camera. So the static camera keeps scanning the horizon and guides the PTZ platform pointing to the latest grid position.

3.4 Multi-class drone classification using deep-learning

After PTZ platform zoom into the region, the view is large enough for operating object detection.

The training model used in the paper is based on an extension of the architecture proposed in [5]. Generally, the region-based object detection can be divided into three sub-networks that perform feature extraction, provide bounding boxes based on the region-proposal network, and perform final classification and bounding box regression. The feature extractor used by this paper is ResNet 101, as proposed in [12].

The RFCN places 100 layers in the shared subnetwork to transfer the time-consuming convolutional neural network to the first two shared subnetworks and uses only one convolution layer to calculate predictions. For dimensionality reduction purposes, the last 1000 class fully connected layer in ResNet101 is replaced by a $1 * 1$ convolution layer with a depth of 1024. The $K^2(C + 1)$ -channel convolutional layer is then used to present the position-sensitive score map. RFCN has proposed the position-sensitive score map to improve the accuracy. The region of interest (RoI) is divided into grids of $K * K$ with a depth of $(C + 1)$ (1 corresponds to the background) after getting region proposals. Every grid has its own scores. Finally, the grid scores are combined to obtain the final output. The combined vote of the scores for each grid contributes to the determination in the respective RoI of the final scores of the $C + 1$ classes. C is the number of classes. Since object detection is trained to identify drones, compared to other objects that may appear in the sky (e.g. birds, aeroplanes, etc.). In this paper C is set to 3, representing drones, birds, and airplanes, respectively.

RFCN's loss function is as follows, which includes loss of classification and loss of regression. As balance weight, the λ equals 1.

$$L(s, t_{x,y,w,h}) = L_{cls}(s_{c^*}) + \lambda[c^* > 0]L_{reg}(t, t^*) \quad (6)$$

In the equation above, c^* is the RoI ground truth label ($c^* = 0$ for background). λ is the balanced weight that is set to 1. Classification uses the cross-entropy loss of $L_{cls}(s_{c^*}) = -\log(s_{c^*})$. The bounding box coordinates are t_x, y, w, h . L_{reg} is a bounding box regression. To train the model, we used annotated images of drones, airplanes, and birds, each with 5000 images of each type. The model is exported in order to make predictions after 200,000 training steps.

If the target is detected as a drone, the system will send an alarm based on the estimated distance between the target and the camera platform and keep tracking the drone until the system mitigates the intruder drone. Otherwise, the system will ignore this target and start searching for a new target (Fig. 7).

3.5 Tracking interface with sensing equipment

To achieve accurate and real-time visual tracking of UAVs, three main challenges need to be addressed, namely (i) computation time, (ii) motion blur, changes in the appearance of the UAV and changes in lighting due to environmental effects, and (iii) drifting between the object and the boundary frame. To address these challenges, the novelty of the proposed framework is the integration of the object detection component described in Section 3.4 with an object tracking algorithm, called Kernelised Correlation Filters (KCF), as presented in [14]. The implementation of the algorithm is outlined in Algorithm 1. The tracker will update the local area of the UAV based on the previous frame (I_{i-1}) and the bounding box (B) of the previously detected UAV. However, if the detection algorithm identifies the UAV in each fifth frame (k) will re-initialize the tracker in the next frame ($k + 1$), based on this frame (k). If the drone is not detected by the detection algorithm, the tracker will be updated continuously based on the previous frame, based on the previous tracking feature. The empirical choice to update the tracker every 5 frames is to balance the detection algorithm's computational complexity with the need to correct for drift caused by the UAV's rapid motion. If the update interval is more than 5 frames, the drift introduced by the KCF could lead to an inability to track the UAV accurately. On the other hand, if the update interval is less than 5 frames, then the detector's computational complexity is too high to enable the UAV to be visually locked in real time.

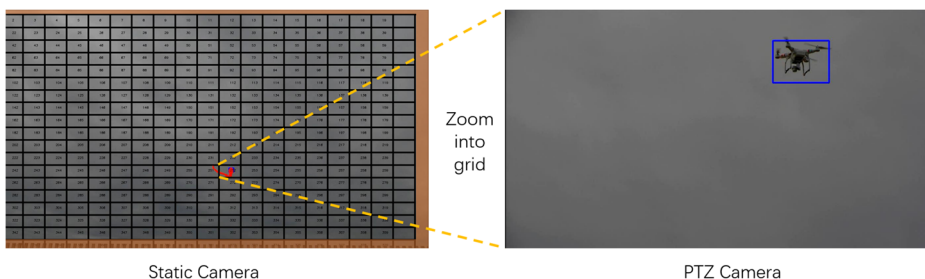


Fig. 7 The result of the drone detection following the PTZ operation using RFCN network

Algorithm 1 Pseudocode for the video analytics algorithm for drone detection.

```

Input: PTZ_video_frame  $I_i$ , Static_video_frame  $I_{s_i}$ 
Result: Drone detected in the horizon  $B$ 
create_tracker;
init_detector;
camera_alignment( $I_0, I_{s0}$ );
do in parallel
  while true do
    Grid_ID=intruder_detection( $I_{s_i}$ );
    initiate_servo_position( $Grid\_ID$ );
  end
end
do in parallel
  if  $i == 1$  then
     $B = detection\_algorithm(I_i)$ ;
     $start\_tracker = false$ ;
     $previous\_frame = I_i$ ;
  else
    while true do
      read_current_frame ( $I_i$ );
      if  $start\_tracker = true$  then
        reinitialise_tracker( $I_{i-1}, B_{i-1}$ );
         $B = calculate\_drone\_position(I_i)$ ;
      else
         $B = update\_drone\_position(I_i)$ ;
      end
      if  $i \% 5 == 0$  then
         $B = detection\_algorithm(I_i)$ ;
         $start\_tracker = true$ ;
      end
      initiate_servo_position( $B$ )
    end
  end
end

```

The results of the detection and tracking algorithm are input to the sensing platform, which controls the movement of the pan-tilt-zoom parameters of the PTZ platform. The relative pan-tilt-zoom parameters are calculated via the *initiate_servo_position* function to trigger the servo motor controller. Based on the captured video footage, the position is determined by the coordinates of the center point of the detected drone B . If the ratio of x coordinates is less than 0.4 or more than 0.6, the platform is triggered to turn left or right. If the ratio of y coordinates is less than 0.3 or greater than 0.7, the platform will be tilted up or down. The scaling parameter is determined by dividing the frame width by the box width. Whether the parameter is greater than 7 or less than 4, this will cause the camera to zoom in or out.

3.6 Alert component

The final component of the proposed framework includes the alarm system to record the drone detection event into the centralised command centre deployed in the cloud for enabling the human operator to trigger a suitable mitigation action. The report metadata contains two key data elements namely (i) event log and (ii) media log. The event log is a JavaScript Object Notation (JSON) structure consisting of detector deployment specification in terms of the orientation of the camera, Geolocation of the detector deployment, the associated reference to the critical infrastructure and the coordinate translation of the detected drone from 2D visual plane to the 3D world coordinate. In addition, to the event log, it is vital to transmitting the associated media element to ensure the human operator is provided with associated evidence prior to the deployment of the respective intruder drone mitigation strategy. While there exist several techniques to transmit continuous media streams through the Internet Protocol (IP) network, the proposed framework adopts the event-emitter design to optimise the data stream transmitted through the network. This is achieved using the event message protocol supported by Apache Kafka [21]. The detector is configured to generate an event for each media timestamp in which the intruder drone is detected. Upon such an event being received, the media bitstream is encoded as a JPEG image and the data is posted to the pre-determined message queue. The cloud-hosted Apache Kafka is interfaced with a consumer, which is hosted upon the Flask server [11]. The consumer end-point is accessed via the web browser to facilitate the visualisation of the media evidence that is associated with the event log. The Uniform Resource Locator (URL) for accessing the media log is also included in the event log for visualising the media object obtained from the Kafka message queues.

4 Experimental result

To evaluate the performance of the whole system, each component tests separately. The algorithm part tested on record videos with ground truth label to evaluate their performance via Average Precision (AP) [7] and Central Distance Curve Metric [3, 29]. The system part tested by field trails in the real world to evaluate the capability of (i) maximum distance for intruder monitor distance by the static camera, (ii) the overall drone detection performance of the dual-camera system and (iii) the performance of the alert component. Those tests are implemented in Olympic Park, London, with DJI Phantom 3 which offer ground truth trajectory with time stamp for evaluating the system.

In the algorithmic part, the proposed security framework for targeting drone attacks is experimentally evaluated using visual analysis techniques in two phases, namely (i) the accuracy of the detection algorithm implemented using RFCN and (ii) the efficiency of the proposed approach in tracking the flight path of the drone. The experimental results are based on a set of video footage in an urban environment taken within 450m from the attack point, using the DJI Phantom 3 Standard as a pilot. A total of 39 attack simulations were created, with the duration of each attack ranging from 6 to 117 seconds, resulting in a total of 12.36 minutes of drone flight.

For the system part, the autonomous drone detection system is installed. The static camera and PTZ camera have the same focal length and imitate the situation of attaching from the horizon. The system is placed on the perimeter of the infrastructure and pointing to the

Table 3 Drone detection accuracy

Method	AP
SSD Mobilenet [15, 17]	30.39%
SSD Inception v2 [27]	7.78%
Faster RCNN Resnet101 [23]	69.49%
RFCN Resnet101	81.16%

direction of outside. When the intruder move towards the infrastructure, the object will be captured by the monitor camera. The long-range distance against which the PTZ operation is triggered. These aspects have been presented in the rest of the Section along with the hyperparameters used in the experimental evaluation. The attack against the infrastructure was simulated using DJI Mavic Pro¹ and Phantom 3² drones, both of which belongs to the class of mini-drones.

4.1 Drone detection accuracy

The evaluation protocols used for drone detection is Average Precision (AP) [7] which is the summary of the shape of the precision/recall curve.

The experimental results are based on a set of video clips piloted from the point of attack, taken by a standard DJI Phantom 3 drone in an urban environment. A total of 39 simulations of attacks were made, with the length of each attack varying from 6 to 117 seconds, resulting in a total flight time of 12.36 minutes for the drone. The performance of the proposed RFCN Resnet101 network was compared with three other models of the deep learning network, namely SSD Mobilenet [17], SSD Inception v2 [27] and Faster RCNN Resnet101 [23]. They have been trained and tested on the same dataset. Result pairs are shown in Table 3.

4.2 Tracking accuracy

The performance of the proposed detection and tracking framework is quantified according to [3, 29], using the performance evaluation of the central distance curve metric for object tracking. The central distance curve metric measures the distance between the center of the tracking bounding box and the center of the ground truth box and summarizes the ratio at different thresholds. The test results of the system shown in Fig. 8 represent the metrics of the central distance curve for different UAV flight paths. The various curves describe the different scenarios in which the UAV flight paths occur. Such scenarios include (i) the drone appearing in front of the sensing platform, (ii) the drone passing in front of the sun, (iii) the drone flight occurring at the horizon, (iv) the drone speed and acceleration exceeding the approved specification due to environmental reality, and (v) the motion aspect added by the sensing equipment while the drone video is being filmed.

A frame is considered to be tracked correctly if the predicted center of the target is within the distance threshold of the ground truth. At lower thresholds, higher accuracy means that the tracker is more accurate, and missing the target will prevent it from achieving perfect accuracy over a very large threshold range. When a representative accuracy score was

¹<https://www.dji.com/uk/mavic>

²<https://www.dji.com/uk/phantom-3-pro>

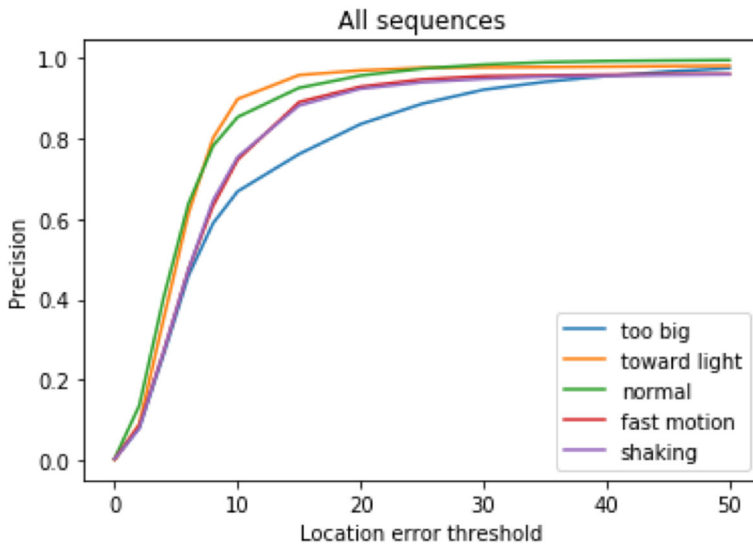


Fig. 8 Central pixel distance curve of different situations

required, a threshold of 20 pixels was selected. In the five scenarios defined for the framework evaluation, the overall average accuracy score of 95.2% for all footages was achieved and is shown in the Fig. 8.

4.3 Intruder detection capability

The capacity of the system depends on the static camera which if operate for intruder detection. The wider and higher resolution camera can cover more area and longer distance. The static camera used in this experiment has the same focal length as the PTZ camera.

The maximum distance between the camera with the drone (Phantom 3) that the intruder detection can detect is shown in Table 4. The ratio is the frame width divided by the bounding box width of intruder object. The motion detector can also be used for intruder detection, which considers the moving object in the scene. However, the maximum distance is too small for attack precaution. Canny with minimum threshold 30 and maximum threshold 100 increase the distance twice as the distance in motion detection. Add some image processing method for preprocessing helps to improve the capability. The frame is sharpened before Canny makes the edge much clearer to detect. After adjusting the threshold of Canny, the maximum distance further extended.

Table 4 The capability regards to the method

Method	Max distance (m)	Ratio
motion detection	80	58
Canny (30,100)	160	116
Canny + sharpen	200	146
Canny (15,300) + sharpen	250	166

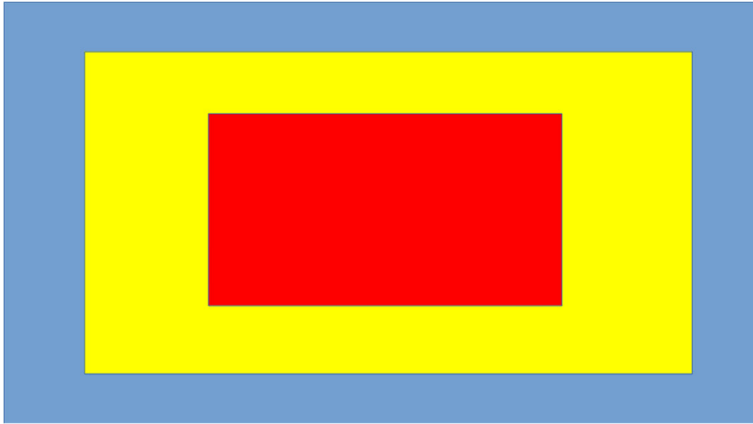


Fig. 9 Performance calculation

4.4 The system detection and tracking performance

The overall functionality of the dual-camera system has two phases: firstly detect the intruder drone from the horizon, then keep tracking the trajectory of the drone’s movement. The detection performance related to Sections 4.1 and 4.3. Since the PTZ camera can zoom into the grid regard to the guidance of static camera, the overall detection part performance has 81.16AP with maximum distance 250m.

Since the goal of the tracking phase is to have the object at the center of the video and we do not have ground truth values in the actual test, we can rate each frame’s output

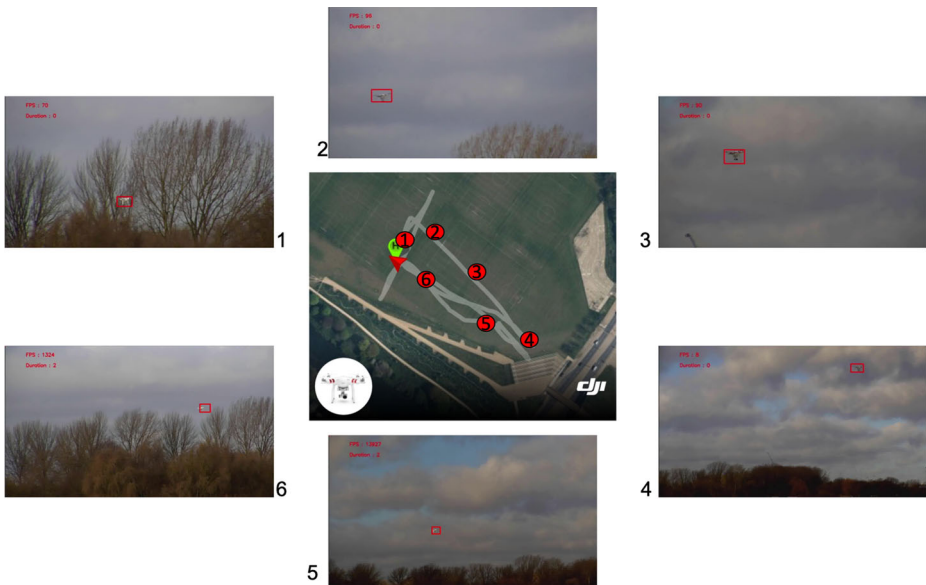


Fig. 10 Trajectory of a drone flight with tracking by PTZ camera

and measure the average score of the total frames. The evaluation method used is shown in Fig. 9. If the bounding box center point is in the red zone, the precision of detection and tracking is 100%. However, if the center point is inside the yellow zone, it is given an accuracy score of 75%. Similarly, if the center is in the blue zone, the frame will be given a 50% accuracy score. Finally, if the region does not contain the drone's presence, a 0% accuracy score is given, resulting in the drone's loss, either through a visual analysis module software implementation or through a camera-controlled delay phase.

The overall pan-tilt-zoom functionality is shown in the Fig. 10, which maps the overall flight path performed by the UAV. In the sense of real-time drone monitoring, one of the main challenges that needs to be tackled is ensuring that the drone is large enough for the video analytics feature to be properly detected. In this regard, it is critical to control and trigger the camera's zoom parameters to check the horizon prior to drone intrusion. In addition, since the momentum and direction of the UAV are heavily influenced by environmental parameters such as wind speed and direction (both headwind and tailwind), the accuracy of the tracking algorithm depends on the latency of the control hardware platform. Thus, three parameters are controlled in the proposed framework, called pan, tilt, and zoom to ensure tracking performance of the UAV. To quantify the sensing platform's performance objectively, it is crucial to consider the feedback received from the component of the analysis. The input given to the analytics component, however, depends on the visual information obtained from the horizon.

4.5 Evaluation of system latency against geographical perimeter

The region is divided into several zones depending on the distance between the object with the location of the camera system as shown in Fig. 11.

For the distance larger than 250m is the safe zone. The object inside this region is ignored which cannot cause any threat to the infrastructure. For the distance between 250m and 150m is the alert zone. Even the distance is still long enough, the object may approaching the infrastructure at a fast speed. So the system pays attention and sends an alarm to the system. For the distance less than 150m is the mitigate zone. The intruder is close enough and can cause damage or attack to the infrastructure so system send strong alarm and trigger other neutralized system to prevent the attack. The field of view is shown in the blue line

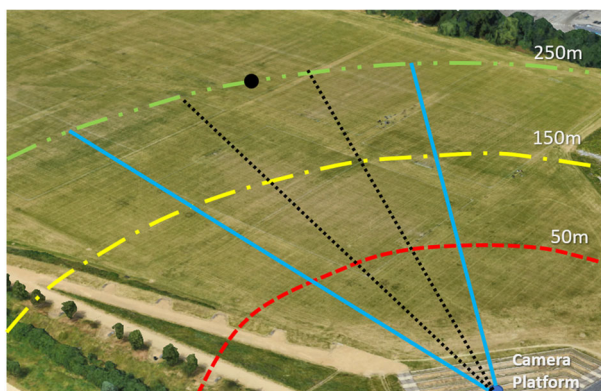


Fig. 11 Protection area

which depends on the focal length of the static camera. The angle in our experiments is 54.94 degree.

In our experiment, the distance estimation is realized by searching in a distance Look-Up-Table(LUT) regards to the size of the detection bounding box and the zoom level of the PTZ camera. We build this LUT in every 10 meters. After getting the estimated distance regarding the protection area definition, the system will send an alarm to show the situation.

5 Discussion

The proposed system has performed sufficiently well detecting drones from 80m to 220m for the attacks simulated by mini-drones, one of the key operational requirement that has been considered out of scope in the paper relates to the distance estimation of the drone upon the detection. While the localisation of the drone position in 3D world coordinate is not possible to be computed without the use of triangulation technique or other suitable approaches, the distance estimation is a vital parameter for the infrastructure operators to be able to develop strategies for the deployment of mitigation protocols. Thus, a brief review of how the visual information captured from the camera could be translated into the distance estimation is presented.

5.1 Distance estimation

In order to locate the intruder drone, the distance between the drone and the camera in real world should be calculated (Fig. 12).

According to the trigonometric relationship with the angle and the edge, the relationship between the real width and the real distance in world coordinate system can be shown as follow:

$$real\ width = 2tan\frac{\alpha}{2} \cdot real\ distance \tag{7}$$

α is the horizontal field of view according to the zoom level, which can be found in Table 2.

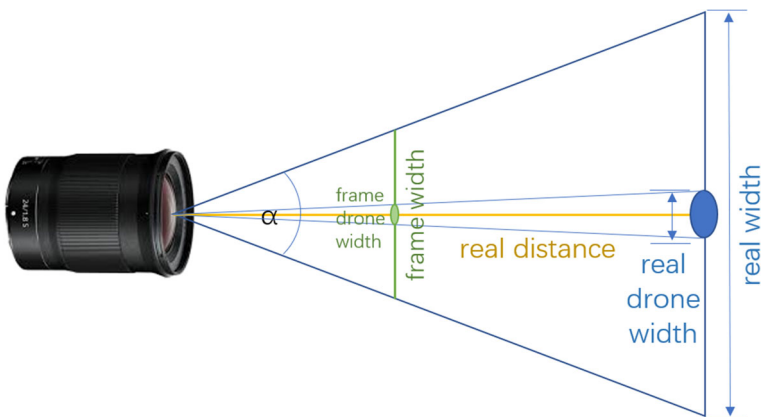


Fig. 12 The relation between the frame and the real world

The ratio of real width of field and the width of the real drone is equal to the ratio of the frame width and the drone width in the frame. So the equation is:

$$\frac{\text{real width}}{\text{real drone width}} = \frac{\text{frame width}}{\text{frame drone width}} \quad (8)$$

And the equation can be transform into:

$$\text{real width} = \frac{\text{frame width}}{\text{frame drone width}} \cdot \text{real drone width} \quad (9)$$

Then put the equation into (7) will get:

$$\text{real distance} = \frac{\frac{\text{frame width}}{\text{drone width in frame}} \cdot \text{real drone width}}{2 \tan \frac{\alpha_{\text{zoom}}}{2}} \quad (10)$$

The frame width equals the resolution of the HD video captured by the camera. The width of the drone detected within the frame is to be obtained by the area of the detected bounding box. The appearance of the drone in the real world is dependent upon the physical dimensions of different drones as specified by the manufacturers. Following the level of focal length parameter of the camera, the distance between the camera and the detected drone can be estimated by (10). The validation of the above mathematical analysis is to be carried out against different sizes of the drone.

6 Conclusion

In this paper, an operational prototype of a long-range drone detection system is presented equipped to protect critical infrastructures against drone attacks. The system presents a seamless operation between the deep-learning algorithm signalling low-cost hardware. The novelty of the paper lies in the ability of the system to detect intruder objects flying at the perimeter of infrastructure with minimal visual information and subsequently attain more visual data to detect malicious drones. The mechanism helps to enlarge the distance of protection. The design of the dual-camera system helps to minimize the latency between the intruder object monitor phase and deep-learning based drone detection phase so that is capable to run in real-time. The combination between RFCN and KCF help to improve the drone detection accuracy and as outlined in the experimental results an overall 95.2% average precision score against drone attacks was achieved by the platform. The future work will continue to investigate approaches to compute the distance which can be used to deploy mitigating strategies for infrastructure security.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Bay H, et al. (2008) Speeded-up robust features (SURF). In: Computer vision and image understanding, vol 110.3, pp 346–359
2. Brust MR, et al. (2018) Defending against intrusion of malicious UAVs with networked UAV defense swarms. In: CoRR arXiv:[1808.06900](https://arxiv.org/abs/1808.06900)
3. Cehovin L, Leonardis A, Kristan M (2015) Visual object tracking performance measures revisited. In: CoRR arXiv:[1502.05803](https://arxiv.org/abs/1502.05803)
4. Cortes C, Vapnik V (1995) Support-vector networks. In: Machine learning, vol 20.3, pp 273–297
5. Dai J, et al. (2016) R-FCN: object detection via region-based fully convolutional networks. In: CoRR arXiv:[1605.06409](https://arxiv.org/abs/1605.06409)
6. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol 1. IEEE, pp 886–893
7. Everingham M, et al. (2010) The pascal visual object classes (VOC) challenge. In: International journal of computer vision, vol 88.2, pp 303–338. ISSN: 1573-1405. <https://doi.org/10.1007/s11263-009-0275-4>
8. Girshick R, et al. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
9. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
10. Gökçe F, et al. (2015) Vision-based detection and distance estimation of micro unmanned aerial vehicles. In: Sensors, vol 15.9, pp 23805–23846
11. Grinberg M (2014) Flask web development: developing web applications with python. 1st. O'Reilly Media, Inc. ISBN: 1449372627, 9781449372620
12. He K, et al. (2015) Deep residual learning for image recognition. In: CoRR arXiv:[1512.03385](https://arxiv.org/abs/1512.03385)
13. He K, et al. (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. In: IEEE transactions on pattern analysis and machine intelligence, vol 37.9, pp 1904–1916
14. Henriques JF, et al. (2014) High-speed tracking with kernelized correlation filters. In: CoRR arXiv:[1404.7584](https://arxiv.org/abs/1404.7584)
15. Howard AG, et al. (2017) MobileNets: efficient convolutional neural networks for mobile vision applications. In: CoRR arXiv:[1704.04861](https://arxiv.org/abs/1704.04861)
16. Li Y, et al. (2019) Scale-aware trident networks for object detection. In: CoRR arXiv:[1901.01892](https://arxiv.org/abs/1901.01892)
17. Liu W, et al. (2015) SSD: single shot multibox detector. In: CoRR arXiv:[1512.02325](https://arxiv.org/abs/1512.02325)
18. Liu W, et al. (2016) Ssd: single shot multibox detector. In: European conference on computer vision. Springer, pp 21–37
19. Lowe DG (1999) Object recognition from local scale-invariant features. In: Proceedings of the seventh IEEE international conference on computer vision, vol 2. IEEE, pp 1150–1157
20. Lu Y, Javidi T, Lazebnik S (2015) Adaptive object detection using adjacency and zoom prediction. In: CoRR arXiv:[1512.07711](https://arxiv.org/abs/1512.07711)
21. Narkhede N, Shapira G, Palino T (2017) Kafka: the definitive guide real-time data and stream processing at scale. 1st. O'Reilly Media, Inc. ISBN: 1491936169, 9781491936160
22. Redmon J, et al. (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
23. Ren S, et al. (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In: CoRR arXiv:[1506.01497](https://arxiv.org/abs/1506.01497)
24. Rozantsev A, Lepetit V, Fua P (2017) Detecting flying objects using a single moving camera. In: IEEE transactions on pattern analysis and machine intelligence, vol 39.5, pp 879–892. ISSN: 1939-3539. <https://doi.org/10.1109/TPAMI.2016.2564408>
25. Singh B, Najibi M, Davis LS (2018) SNIPER: efficient multi-scale training. In: CoRR arXiv:[1805.09300](https://arxiv.org/abs/1805.09300)
26. Sivic J, Zisserman A (2003) Video google: a text retrieval approach to object matching in videos. In: Null. IEEE, p 1470
27. Szegedy C, et al. (2015) Rethinking the inception architecture for computer vision. In: CoRR arXiv:[1512.00567](https://arxiv.org/abs/1512.00567)
28. Taha B, Shoufan A (2019) Machine learning-based drone detection and classification: state-of-the-art in research. In: IEEE access, vol 7, pp 138669–138682. ISSN: 2169-3536. <https://doi.org/10.1109/ACCESS.2019.2942944>
29. Wu Y, Lim J, Yang M (2013) Online object tracking: a benchmark. In: 2013 IEEE conference on computer vision and pattern recognition, pp 2411–2418. <https://doi.org/10.1109/CVPR.2013.312>

30. Yoshihashi R, et al. (2017) Learning multi-frame visual representation for joint detection and tracking of small objects. In: CoRR arXiv:[1709.04666](https://arxiv.org/abs/1709.04666)
31. Zhang X, Chandramouli K (2019) Critical infrastructure security against drone attacks using visual analytics. In: Tzovaras D, et al. (eds) Computer vision systems. Springer International Publishing, Cham, pp 713–722. ISBN: 978-3-030-34995-0

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Xindi Zhang¹  · Kusrini Kusrini²

Kusrini Kusrini
kusrini@amikom.ac.id

- ¹ Multimedia and Vision Group, School of Electronic Engineering and Computer Science, Queen Mary University of London, Mile End Road, London E1 4NS, UK
- ² Universitas AMIKOM Yogyakarta, Jl. Ringroad Utara Condong Catur Depok Sleman, Yogyakarta, Indonesia