

Article

# Real-Time Detection of Ground Objects Based on Unmanned Aerial Vehicle Remote Sensing with Deep Learning: Application in Excavator Detection for Pipeline Safety

Lingxuan Meng <sup>1</sup>, Zhixing Peng <sup>1</sup>, Ji Zhou <sup>1,\*</sup>, Jirong Zhang <sup>1</sup>, Zhenyu Lu <sup>1</sup>, Andreas Baumann <sup>2</sup> and Yan Du <sup>3</sup>

<sup>1</sup> School of Resources and Environment, Center for Information Geoscience, University of Electronic Science and Technology of China, Chengdu 611731, China; 201811070106@std.uestc.edu.cn (L.M.); pengzhixing@std.uestc.edu.cn (Z.P.); zhangjirong@std.uestc.edu.cn (J.Z.); 2016180202023@std.uestc.edu.cn (Z.L.)

<sup>2</sup> Geosensors and Engineering Geodesy, Institute of Geodesy and Photogrammetry, ETH Zürich, 8092 Zürich, Switzerland; andreas.baumann@geod.baug.ethz.ch

<sup>3</sup> Gas Transmission Management Division, PetroChina Southwest Oil & Gasfield Company of CNPC, Chengdu 610051, China; du\_yang@petrochina.com.cn

\* Correspondence: jzhou233@uestc.edu.cn

Received: 7 November 2019; Accepted: 23 December 2019; Published: 3 January 2020

**Abstract:** Unmanned aerial vehicle (UAV) remote sensing and deep learning provide a practical approach to object detection. However, most of the current approaches for processing UAV remote-sensing data cannot carry out object detection in real time for emergencies, such as firefighting. This study proposes a new approach for integrating UAV remote sensing and deep learning for the real-time detection of ground objects. Excavators, which usually threaten pipeline safety, are selected as the target object. A widely used deep-learning algorithm, namely You Only Look Once V3, is first used to train the excavator detection model on a workstation and then deployed on an embedded board that is carried by a UAV. The recall rate of the trained excavator detection model is 99.4%, demonstrating that the trained model has a very high accuracy. Then, the UAV for an excavator detection system (UAV-ED) is further constructed for operational application. UAV-ED is composed of a UAV Control Module, a UAV Module, and a Warning Module. A UAV experiment with different scenarios was conducted to evaluate the performance of the UAV-ED. The whole process from the UAV observation of an excavator to the Warning Module (350 km away from the testing area) receiving the detection results only lasted about 1.15 s. Thus, the UAV-ED system has good performance and would benefit the management of pipeline safety.

**Keywords:** YOLOv3; deep learning; real-time detection; unmanned aerial vehicle (UAV); remote sensing

---

## 1. Introduction

Object detection from remote-sensing images is useful in many fields [1]. The detection of ground objects requires the acquisition and interpretation of remote-sensing images. Unmanned aerial vehicles (UAVs) equipped with global positioning system modules and high-resolution digital cameras are able to provide remote-sensing images and accurate geolocation information [2]. This technique has been demonstrated to be a very effective way to collect data for a large area in a timely, cost-efficient and convenient way. UAV remote sensing plays an important role in many

fields, including photographic measurements [3], environmental protection [4], search and rescue operations [5], precision agriculture [6–9], infrastructure monitoring [10], and traffic management [11–13]. Deep-learning algorithms, which have achieved state-of-the-art performance on a wide range of image processing tasks, are able to interpret the remote-sensing images efficiently. These algorithms overcome the drawbacks of the manual interpretation of remote-sensing images such as time and high economic costs. In recent years, deep learning has been used in many fields for processing UAV remote-sensing images, such as for pedestrian detection [14], land-cover classification [15], ecological protection [16], and digital terrain model (DTM) extraction [17].

The conventional approach for combining UAV remote sensing and deep learning for the detection of ground objects can be divided into two steps [18]. The first step is to capture remote-sensing images during the UAV flight; the second step is to copy these images to the computer and then analyze them on the computer. Many studies have been conducted successfully according to the aforementioned workflow and have proposed a number of approaches to solve the associated problems. For example, in disaster monitoring, a new deep learning-based wildfire identification method was proposed and tested successfully to locate and extract core fire areas [19]. In precision agriculture, a fully automatic method was employed for weed detection using convolutional neuronal networks (CNNs) with an unsupervised training dataset collected from UAV remote-sensing images [20]. The results of this method are comparable to those of traditional, supervised training methods. An advanced method was presented to detect vine diseases in images obtained by UAV and red green blue (RGB) sensors [21]. The proposed method combined the strength of the deep learning approach with different color spaces and vegetation indices to obtain the best results. In livestock censuses, an in-depth study was carried out on a large mammal census in an African savanna wildlife reserve based on UAV images and deep learning [22]. In addition, in the continuous safety monitoring and maintenance of infrastructure, a commercial UAV with a digital camera and convolutional neural networks was used to identify cracks in an aging concrete bridge [23]. These studies demonstrate that deep learning has good capabilities in processing UAV remote-sensing images.

Nevertheless, in many applications, such as fire-fighting, forest fire prevention, pipeline monitoring, precision architecture, and search and rescue, it is necessary to detect the ground objects in UAV images in real time and transfer the related information to the users as quickly as possible. For example, oil and gas pipelines are threatened constantly by construction machinery such as excavators located close by. Reports showed that in the past 20 years, more than 700 pipeline accidents occurred across the United States. These accidents caused more than 250 fatalities and excavation was the leading cause of pipeline breaks [24]. Thus, the real-time monitoring of excavators is an important issue in preventing pipeline accidents. According to the law on the protection of petroleum and natural gas pipelines in China, no excavator is allowed to work within a five meters safety margin on both sides of a pipeline. When an excavator appears in this region, it increases the risk of a pipeline accident; in this situation, pipeline safety managers have to response immediately. The conventional approach for excavator monitoring is carried out in the field by laborers: workers walk along the pipeline to record the status of the pipeline and prevent behaviors that threaten the safety of the pipeline. However, this approach is inefficient and incurs high labor costs. In addition, installing cameras beside pipelines can allow constant pipeline monitoring, but only a small section of pipeline can be monitored. To cover the entire pipeline, many cameras would need to be installed along the pipeline. Considering the costs associated with camera installation and maintenance, the power supply, and the Internet supply, it is impossible to monitor the entire pipeline in this way [25]. In practical situations, only the key parts of the pipeline are equipped with fixed cameras for constant monitoring. Although there are a few applications using UAV remote sensing to monitor excavator activity along a pipeline, the conventional work flow for processing UAV remote-sensing data cannot satisfy the real-time monitoring of excavators.

Currently, there are some studies reporting the real-time processing of UAV remote-sensing data. A few of them have focused on algorithms for the real-time processing of UAV images on a computer, especially in the field of object detection [26,27]. Some studies have used commercial

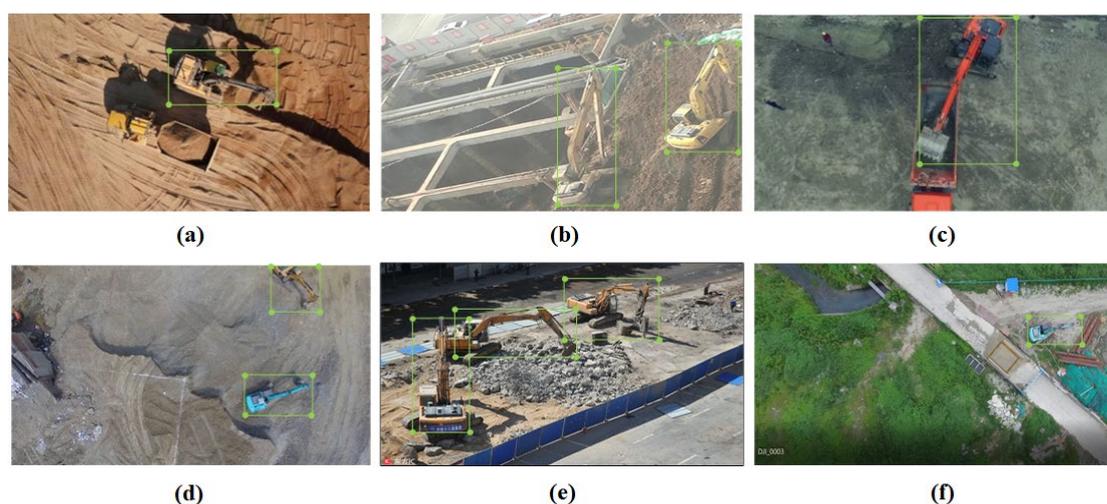
image transmission modules to transfer the UAV remote-sensing data to the computer and process the data on the computer in order to realize the real-time processing of the data [28,29]. This approach may suffer from the unstable and short-distance transmission of the transmission module in practical applications. In some other studies, an embedded board is integrated on the UAV as the processing unit, such as the NVIDIA Jetson TX2 board, the Jetson AGX Xavier board, and the Intel Neural Compute Stick [30,31]. This approach ensures that the UAV data is processed stably and in real time. Unfortunately, studies on the real-time processing of UAV data on the UAV itself are still rare. Additional comprehensive studies are needed to provide more references for the community. In industry, some commercial applications have made use of real-time processing to detect and track targets, such as the Skydio R1 [32] and the DJI Mavic 2 [33]. Recently, the Little Ripper company and the University of Technology Sydney developed a system to enhance beach safety; this system is able to monitor the activity of sharks and send real-time video streams to the UAV pilots on the beach [34]. However, the promotion of these commercial applications is limited because their data transmission distance is restricted by the transmission module. Therefore, in order to realize the real-time detection of ground objects based on UAV remote sensing, some key functional parts need to be implemented and integrated, including a fast and accurate object detection algorithm, integration of the algorithm, UAV, and camera, and the transmission of the processing results to the users.

In this context, this study proposes an approach for realizing the real-time detection of ground objects by integrating UAV remote sensing and deep learning. Furthermore, a system is constructed for operational application. Excavators, which greatly threaten the safety of pipelines, are selected as the ground objects to be detected. Thus, the constructed system is termed UAV-ED (i.e., UAV for excavator detection). Several experiments in different scenarios are conducted to evaluate the performance of the proposed approach and UAV-ED system. This study would be beneficial for preventing pipeline accidents and similar applications in other fields.

## 2. Materials

### 2.1. Dataset

To train the deep-learning model, 350 images containing excavators were collected. Seventy percent of these images were obtained from the Internet and the remainder were obtained by acquiring UAV images at construction sites. Images without excavators were not collected. Because excavators often appear at construction sites, about 15% of collected images contained other construction equipment besides excavators, such as grab machines and cranes, which were not the detection target. About 60% of collected images contained multiple excavators in one image. We annotated every image by drawing bounding boxes around each excavator manually with the labeling graphical image annotation tool [35]. Examples of the annotated images are shown in Figure 1. The images collected from the Internet were different in illumination, background and resolution; and the excavators contained in them are different in size, shape, color, and posture to ensure the robustness of the excavator detection model. Furthermore, a UAV equipped with a electro-optical (EO) camera was used to take videos at three construction sites near the University of Electronic Science and Technology of China (UESTC), with a flight height ranging from 100 m to 200 m. These videos were obtained in both vertical view and oblique views. Each video was taken with a size of  $1920 \times 1080$  pixels. We extracted the images containing excavators in different postures and backgrounds from the videos. All 350 images were combined and shuffled to form the dataset for the subsequent excavator detection model.



**Figure 1.** Examples of the annotated images containing excavators of different sizes, colors, and postures. The images were taken from different viewing directions. (a) Vertically viewed image with a yellow excavator; (b) obliquely viewed image with two yellow excavators; (c) obliquely viewed image with a red excavator; (d) vertically viewed image with two excavators of different colors; (e) obliquely viewed image with three yellow excavators in different postures; and (f) vertically viewed image with a blue excavator.

## 2.2. Instruments

As shown in Table 1, the instruments in this study included an UAV, a sensor (i.e., EO camera), an embedded board, a tablet, and a computer. We used a UAV equipped with a high-resolution EO camera to obtain the aerial images. The embedded board is used to deploy deep learning algorithms to process the captured images in real time. The tablet and the computer are used to display the detection result to the users.

**Table 1.** Instruments used in this study for excavator detection.

Instrument	Model	Specification
Unmanned aerial vehicle (UAV)	DJI M600 Pro	Six rotors
		Image transmission module: DJI LightBridge 2 Flight time: 38 min with no load Maximum horizontal velocity: 18 m/s Load capacity: maximum 6 kg
Sensor	FLIR BFLY-U3-13S2C-CS	Resolution: 1288 × 964 pixels
		Size: 29 × 29 × 30 mm
		Weight: 36 g
Embedded board	NVIDIA Jetson TX2	Frame rate: 30 frames per second
		Graphics Processing Unit (GPU): an NVIDIA Pascal™-family GPU
		Memory: 8 GB
		Memory bandwidth: 59.7 GB/s
		Power consumption: 7.5 watt
		Support hardware interfaces: Universal Serial Bus and High Definition Multimedia Interface
Tablet	Apple iPad mini 4	-
Computer	Microsoft Surface Pro 4	-

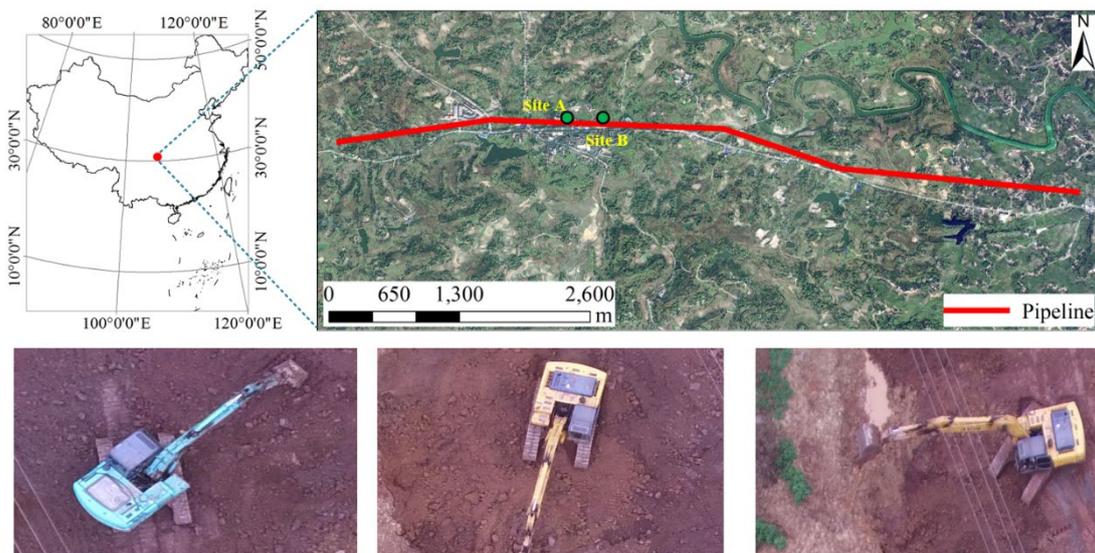
The UAV platform used in this study was a DJI Matrice 600 (M600) Pro because of its good support for secondary development and its high stability. DJI M600 Pro is equipped with DJI LightBridge 2 transmission system to meet the requirement of high-definition live streaming. The image signal transmission distance is as far as 5 km in unobstructed areas. The sensor used in this study was a FLIR industrial camera BFLY-U3-13S2C-CS. This RGB camera is a low-cost, commercial, and off-the-shelf product with the advantage of low power consumption and high resolution. Furthermore, it is light weight and has a good stabilization sensor to reduce blurring of none-stable platforms, which is an advantage for use on drones. The camera's focal length supports manual adjustment to adapt multiple flight altitudes.

The embedded board used in this study was a NVIDIA Jetson TX2 board, one of the fastest, most power-efficient embedded artificial intelligence (AI) computing devices. It is built around an NVIDIA Pascal™-family GPU and loaded with 8 GB of memory and 59.7 GB/s of memory bandwidth. It features a variety of standard hardware interfaces that make it easy to be integrated into a wide range of products.

The study also used the OneNET platform [36] to realize the remote, wireless, and real-time transmission required by the UAV-ED system. The OneNET platform is an open and free cloud platform created by China Mobile, that simplifies the access of Internet of Things (IoT) devices to the cloud. The OneNET platform has already been applied in various business fields, such as environmental monitoring and precision agriculture [37].

### 2.3. Unmanned Aerial Vehicle (UAV) Remote-Sensing Experiment

The area for testing the proposed approach and UAV-ED system is located in Qu county, Sichuan province, China (Figure 2). The length of the pipeline in Qu county is approximately 35 km. Due to the abundant soil resources and many infrastructure constructions projects in Qu county, excavators often work around the pipeline and, thus, seriously threaten the pipeline's safety. Currently, the pipeline is inspected in the field by laborers twice a day. Because the pipeline is 20 to 100 m away from the main road, the manual monitoring approach is inefficient and time-consuming. Due to the limitation of the flight distance of the UAV, a part of the pipeline was selected to test the proposed approach and UAV-ED system. The total length of the selected pipeline was about 5 km. The land-cover types of the selected area included grassland, bare soil, built-up surface, ponds, and roads. Therefore, the selected area was very typical for testing the proposed approach and UAV-ED system.



**Figure 2.** Area for testing the proposed approach and UAV for excavator detection (UAV-ED) system and the images with excavators taken by the UAV in the experiment. The green solid points in the map denote the sites where the excavators were working.

The UAV remote-sensing experiment was conducted on 15 September 2018. The weather was cloudy all day. We found there were excavators working on two sites (Figure 2). From 10:00 am to 6:00 pm, 5 flights were conducted, once every two hours. The DJI M600 Pro UAV, equipped with a FLIR EO camera and a Jetson TX2 embedded board, was employed. The duration of each flight was 25–30 min. Three of the five experiments were performed along the pipeline by automatic control through the UAV ground control station, and another two experiments around the excavators were performed by manual control with a UAV external pilot. In the three flight missions, the flight speed of the pipeline monitor ranged from 5 m/s to 10 m/s, the flight altitude ranged from 100 m to 200 m, and the maximum flight distance was about 5 km. The FLIR camera on the UAV was installed vertically and oriented to the ground during pipeline monitoring; thus, the captured images were top views of the ground objects.

### 3. Methodology

#### 3.1. The Excavator Detection Model

Many neural network models for object detection have been proposed and have exhibited good performance. These models are usually split into two categories according to their architecture: two-stages detectors and one-stage detectors. For the two-stages detectors, such as R-CNN [38], Fast R-CNN [39], and Faster R-CNN [40], a region proposal network is used to generate regions of interest in the first stage. In the second stage, the objection classification and bounding box regression are performed. Although two-stage detectors generally yield better accuracy than one-stage detectors, they are not applicable for the real-time detection of ground objects due to their high computational costs.

Compared with two-stage detectors, one-stage detectors, such as You Only Look Once [41] and Single Shot MultiBox Detector [42], do not require the first-step for region proposal. They treat object detection as a simple regression problem. Therefore, one-stage detectors are generally faster than two-stage detectors. In addition, the accuracy of one-stage detectors is acceptable in many applications [23,43]. The You Only Look Once V3 (YOLOv3) model [43] is one of the most widely used one-stage detectors [38,44]. Its architecture consists of two parts, a feature extractor and an object detector. Compared with two-stage detectors, the YOLOv3 model has several advantages. It extracts the features under the global context of the entire image, and it is very fast since it only needs one stage. Considering the processing speed for real-time detection of the excavators, the computational costs, and the accuracy in detection, the YOLOv3 model was used in this study.

The 350 images with excavators described in Section 2 were used to train the excavator detection model based on YOLOv3. Since the number of the images is not very large, the trained model may suffer from overfitting. Thus, the excavator detection model was trained with a transfer learning strategy based on the pretrained YOLOv3 model published by [43]. One should note that the pretrained YOLOv3 model was trained with ImageNet, which is a large visual database designed for visual object recognition. However, images of ImageNet were not obtained from the aerial viewpoints. Thus, the pretrained YOLOv3 model was further trained based on the 350 collected images, with 80% of the images for training, 10% for validating and 10% for testing. The transfer learning strategy can also significantly decrease the time consumed in the training process.

In the training process, five parameters, including the recall rate, accuracy, precision, intersection over union (IoU), and loss function, were selected to evaluate the model's performance. The recall rate, accuracy and precision were used to measure the ratio of excavators detected by the excavator detection model in the images. IoU was used to measure the ratio of overlap between the real excavator bounding box labeled manually and the predicted excavator bounding box given by the excavator detection model. Loss function was used to evaluate the overall performance of the excavator detection model from the location accuracy and detection accuracy. The formulas for these three parameters are listed below:

$$\text{Recall} = \frac{\text{TP}}{\text{FN} + \text{TP}} \quad (1)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{FN} + \text{TP} + \text{FP} + \text{TN}} \quad (3)$$

where TP (true positives) denotes the number of the true excavators that are correctly identified by the excavator detection model; and FN (false negatives) denotes the number of true excavators that are falsely identified as background by the model; TN (true negatives) denotes the number of the background that are correctly identified by the excavator detection model; and FP (false positives) denotes the number of the background that are falsely identified as excavators by the excavator detection model. Because only the excavators in images are labeled and background is not the detection target in our experiment, TN is ignored. Accuracy can be calculated by recall rate and precision.

$$\text{IoU} = \frac{\text{AoO}}{\text{AoU}} \quad (4)$$

where AoO is the area of overlap between the real excavator bounding box labeled manually and the predicted excavator bounding box given by the excavator detection model; and AoU is the area of union between the real excavator bounding box labeled manually and the predicted excavator bounding box given by the excavator detection model.

Loss function consists of four terms:

$$\text{Loss} = E_{xy} + E_{wh} + E_{\text{confidence}} + E_{\text{class}} \quad (5)$$

where  $E_{xy}$  is the loss of the location of the center of the predicted object;  $E_{wh}$  is the loss of width and height of the predicted bounding box;  $E_{\text{confidence}}$  is the loss of confidence of the predicted object; and  $E_{\text{class}}$  is the loss between the predicted category and the real category. Since the excavator is the only class to be detected in the detection model,  $E_{\text{class}}$  is equal to 0.

The architecture of the excavator detection model based on YOLOv3 is shown in Figure 3. It is designed in [43]. It has 75 convolutional layers, with residual layers and upsampling layers. The convolutional layers produce feature maps by convolving the input image with a specified number of kernels. The convolutional layers with stride 2 are used to down sample the feature maps, which reduce the loss of the low-level features compared with using the pooling layers. For the convolutional layers with the kernel size of  $3 \times 3$ , the convolution process is carried out as follows:

$$O_{w,h,d} = \left( \sum_{i=-1}^1 \sum_{j=-1}^1 \sum_{k=1}^c W_{i,j,k}^d I_{w+i,h+j,k} \right) + b^d \quad (6)$$

where  $I_{w+i,h+j,k}$  is the input,  $O_{w,h,d}$  is the output, and  $W_{i,j,k}^d$  and  $b^d$  are the weights and the bias of the  $d$ 'th kernel of the convolution layer, respectively. The number of kernels is the same as the number of channels of the output feature maps.  $c$  is the number of channels of the input feature maps. As  $k$  takes 1, ...,  $c$ , each kernel of the convolution layer works on the entire channels of the input feature maps. The convolution process of the convolutional layer with the kernel size of  $1 \times 1$  is similar to that of  $3 \times 3$ . An activation function, the leaky relu function, is used to perform non-linear transformation on each output of the convolution, which enables the model to learn more complex features. The leaky relu function is listed below:

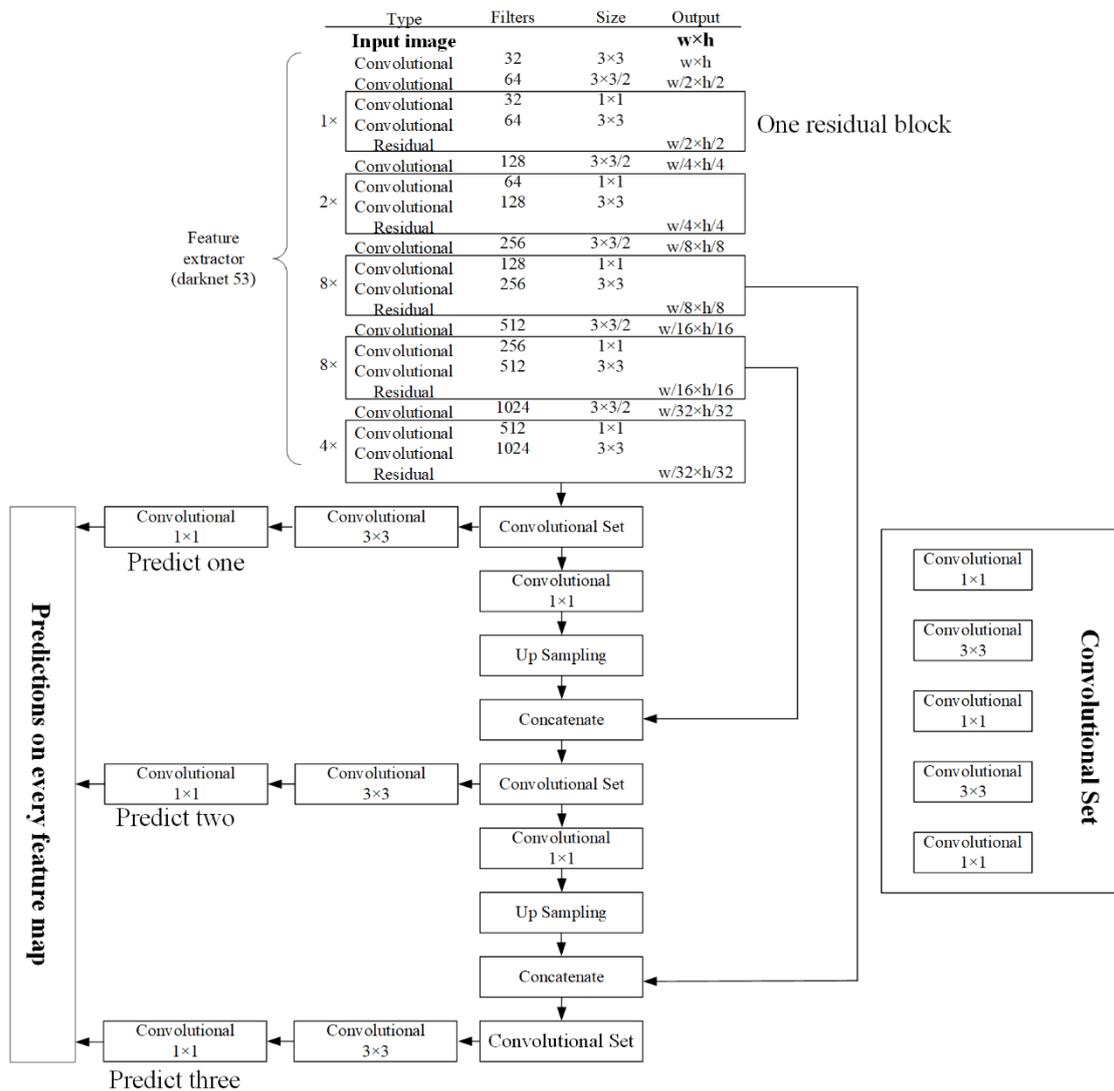
$$O_{w,h,d} = \begin{cases} O_{w,h,d} & \text{if } O_{w,h,d} \geq 0 \\ 0.1 \times O_{w,h,d} & \text{if } O_{w,h,d} < 0 \end{cases} \quad (7)$$

Residual blocks are used to address the degradation problem by learning the residual features instead of directly learning the underlying features [36]. As depicted in Equation (6), the model gets the desired features  $O_{w,h,d}$  by adding the input of the residual block  $[O-3]_{w,h,d}$  and the output of the stacked layers of the residual block  $[O-1]_{w,h,d}$  (i.e., the residual features).

$$O_{w,h,d} = [O-1]_{w,h,d} + [O-3]_{w,h,d} \tag{8}$$

The nearest neighbor algorithm was used to upsample the feature map from the size of  $w/32 \times h/32$ ,  $w/16 \times h/16$  to  $w/16 \times h/16$ ,  $w/8 \times h/8$ . The excavator detection model makes predictions on three feature maps by the convolutional layer with the kernel size of  $1 \times 1$  and the sigmoid function. The sigmoid function is listed as follows:

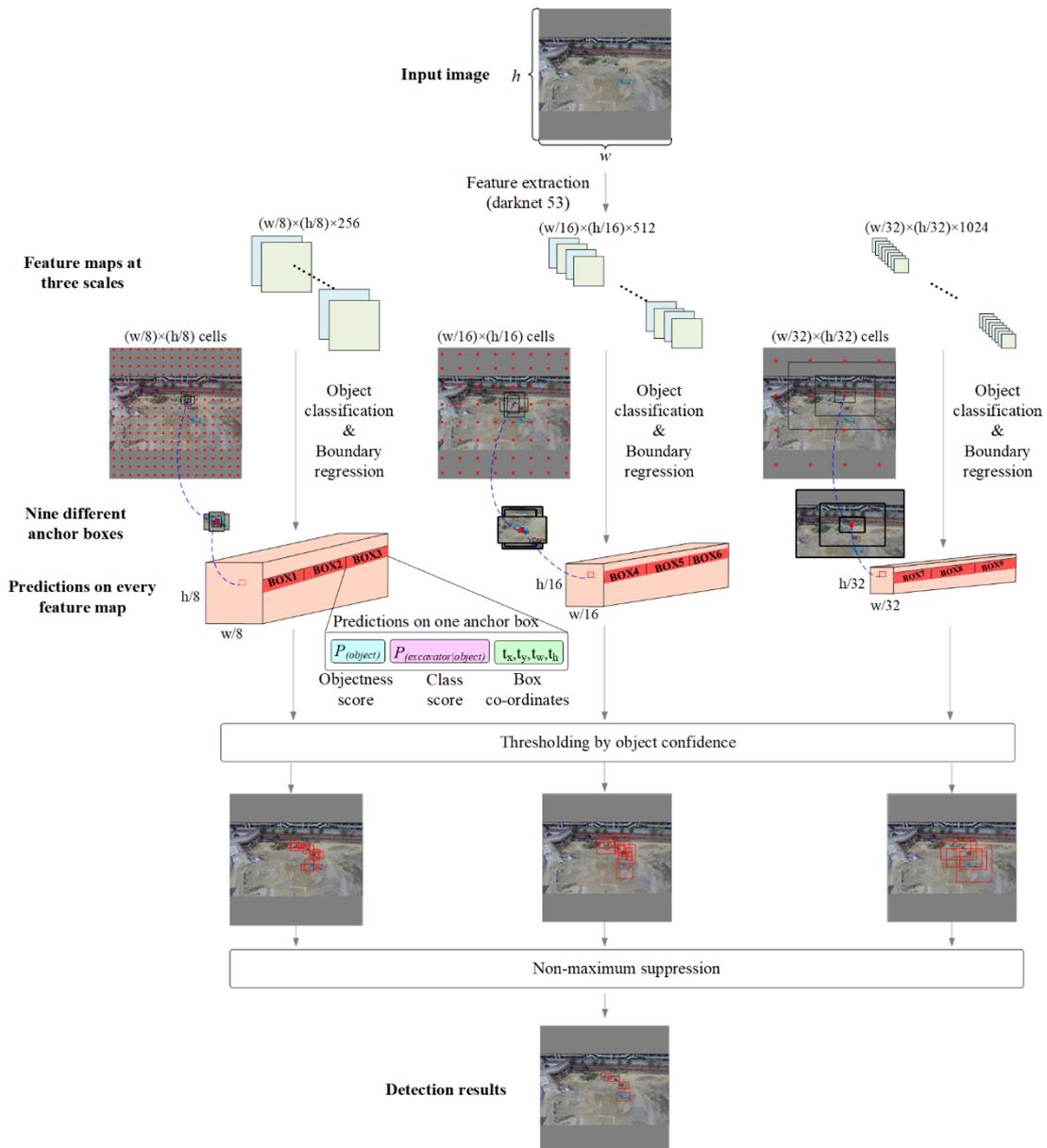
$$S(x) = \frac{1}{1 + e^{-x}} \tag{9}$$



**Figure 3.** The architecture of the excavator detection model based on You Only Look Once V3 (YOLOv3). It has 75 convolutional layers, with residual layers and upsampling layers. The excavator detection model uses darknet53 as the backbone to extract the features of the input image and detects the excavators on three different scale feature maps [43].

The flowchart of the trained excavator detection model is shown in Figure 4. During the detection process, the darknet53 network model [45] under YOLOv3 was employed to extract the features from the entire image and generate feature maps at three scales. The three feature maps

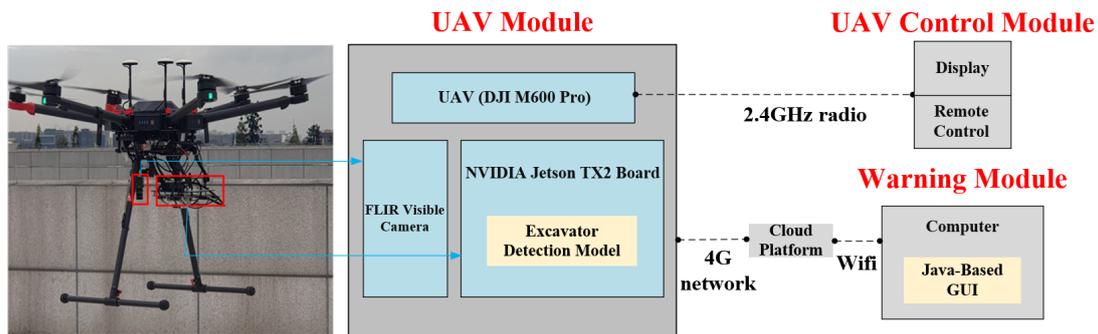
have  $(w/8) \times (h/8)$ ,  $(w/16) \times (h/16)$ , and  $(w/32) \times (h/32)$  cells, respectively ( $w$  is the width of the image and  $h$  is the height). Cells in different feature maps have different receptive field sizes [37]. There are nine different anchor boxes used to make predictions in the three feature maps in total. For each feature map, object classification is performed in every cell to calculate the confidence of containing excavators at three different anchor boxes with the cell as the center. In addition, the three anchor boxes are used to perform the boundary regression to predict the bounding box of the excavator for each cell. The predicted outputs on one anchor box of one cell contain six values, including the center coordinate of the bounding box  $(tx, ty)$ , the width and height of the bounding box  $(tw, th)$ , the confidence for the anchor box containing the target object  $(P(\text{object}))$ , and the confidence of the detected object being an excavator  $(P(\text{excavator}|\text{object}))$ . Since excavators are the only target object in this experiment, the confidence of the detected object being an excavator is always 100%. Using three scales to extract features and nine anchor boxes to perform the boundary regression makes the trained excavator detection model able to detect excavators of different sizes and postures. To prevent the detection of the same excavator several times, the confidence obtained and the predicted bounding box are subjected to objectness score thresholding and non-maximum suppression (NMS) to obtain the most likely bounding box of the detected excavator.



**Figure 4.** Flowchart of the excavator detection model. The red dot is the center of the receptive field for each cell on the original image. The red boxes are the predicted bounding boxes with confidence beyond the object score threshold (i.e., 0.5). Among them, the red boxes in the detection results are the most likely bounding boxes of the detected excavators.

### 3.2. Construction of the UAV for Excavator Detection (UAV-ED) System

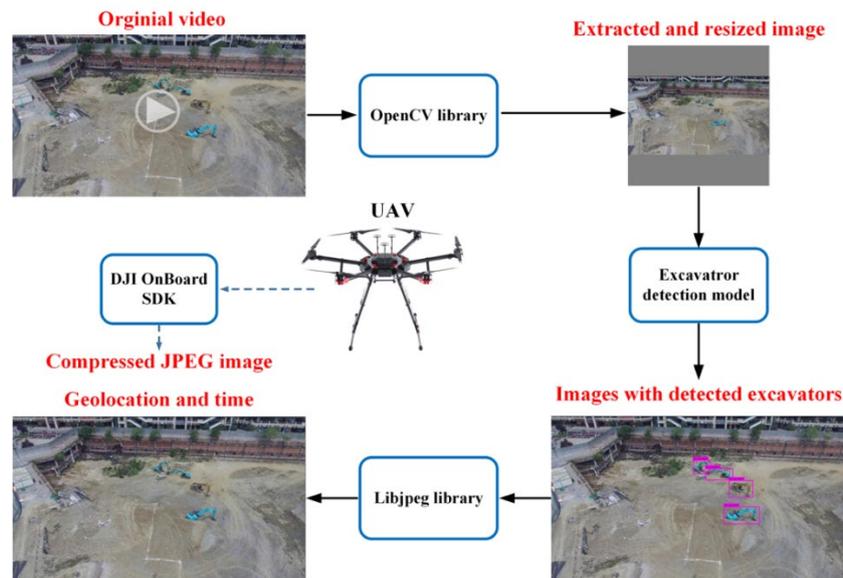
Integration of the aforementioned excavator detection model, the UAV, and related components makes it possible to establish a system to monitor the excavators automatically. The UAV-ED system is composed of three modules, including the UAV Module, the UAV Control Module, and the Warning Module. The architecture of this system is shown in Figure 5. Details of these modules are presented in the following subsections.



**Figure 5.** Architecture of the developed UAV-ED system. The UAV-ED system is composed of three modules, including the UAV Module, the UAV Control Module, and the Warning Module. The UAV Module integrated the excavator detection model, the sensor and the UAV. The UAV Control Module and the Warning Module are used to realize the real-time transmission of the detected excavators' information.

#### 3.2.1. Integration of the Excavator Detection Model, Sensor and UAV

The UAV Module is to detect the excavator, determine the geolocation, and transfer the information to the other two modules. The main hardware components of the UAV module are the UAV (DJI M600 Pro), the EO camera (FLIR BFLY-U3-13S2C-CS), and the embedded board (NVIDIA Jetson TX2 board). In addition, the UAV has a mounting board to mount the embedded board and a gimbal to mount the FLIR camera. Before mounting the NVIDIA embedded board on the UAV, the previously trained excavator detection model was deployed in the embedded board. In addition, three libraries were imported into the embedded board, including the Open Source Computer Vision (OpenCV) library in version 3.4, the Libjpeg library, and the DJI onboard Software Development Kit (SDK). OpenCV is an open source computer vision and machine learning software library, which was built to provide a common infrastructure for computer vision applications [46] [47]. The Libjpeg is a widely used C library for reading and writing JPEG image files [48]. The DJI Onboard SDK is an open source software library that enables computers to communicate directly with DJI UAVs and provides access to the UAVs' telemetry [49]. The data-processing flow and the relationship between the excavator detection model and the three previously mentioned libraries are shown in Figure 6.



**Figure 6.** Data-processing flow in the UAV Module. Images are extracted from the original video and resized by the OpenCV library. Then, each image is orderly imported into the excavator detection model to identify if there are any excavators in the image. Images containing excavators are compressed via the Libjpeg library and transmitted to the Warning Module along with the recorded geolocation and acquisition time.

The OpenCV library is used to extract images from the video acquired by the camera. The extracted images have the same resolution as the video, namely,  $1288 \times 964$  pixels. The images are resized simultaneously to facilitate the processing of the aforementioned excavator detection model. In the process of image extraction and resizing, approximately 5.9 images with a size of  $288 \times 288$  pixels or 3.7 images with a size of  $416 \times 416$  pixels can be obtained per second. Afterward, each image is orderly imported into the excavator detection model to identify whether there are any excavators in the image.

If an excavator is detected on the resized image, it is marked by a box through the OpenCV library on the corresponding image before resizing. The images are compressed into the JPEG format via the Libjpeg library to facilitate the transmission of the images with excavators to the Warning Module. In addition, the system reads the geolocation (i.e., longitude and latitude) and time from the flight controller of the UAV with the DJI Onboard SDK. The compressed JPEG image, the recorded geolocation, and the acquisition time are transmitted to the Warning Module. One should note that the camera is mounted vertically facing the ground; thus, the geolocation of the UAV is treated closed to that of the detected excavator.

### 3.2.2. Real-Time Transmission of the Detected Excavators' Information

It is required that the users obtain the detected excavators' information in a timely and convenient manner to take corresponding actions and ensure the pipeline's safety. Thus, the real-time transmission of the associated information is crucial. Real-time transmission is realized both in the UAV Control Module and Warning Module. The instruments of the Control Module are a tablet and a UAV remote-control unit. With DJI Go App and DJI GS Pro App installed on the tablet, users can set the flight missions of the UAV and control its flight state. In addition, the video formed by the orderly extracted images (e.g., 5.9 images with a size of  $288 \times 288$  pixels per second) can be displayed on the tablet. In this newly generated video, users can see detected excavators within boxes generated by the OpenCV library.

For the Warning Module, software with a Java-based Graphical User Interface (GUI) is developed and run on a computer connected with the Internet. The software is able to receive a compressed JPEG image, the geolocation, and time immediately an excavator is detected by the UAV Module. The transmission between the UAV Module and the Warning Module is remote,

wireless, in real-time, and realized through the OneNET cloud platform. The UAV Module uploads the data package (namely, the compressed JPEG image, the geolocation, and time) to the cloud platform through the 4G network; then the cloud platform sends the data package immediately to the software of the Warning Module. The geolocation is also linked to the digital map. Based on the image with the excavators, the geolocation, and the time, the manager of the pipeline can respond rapidly.

## 4. Results

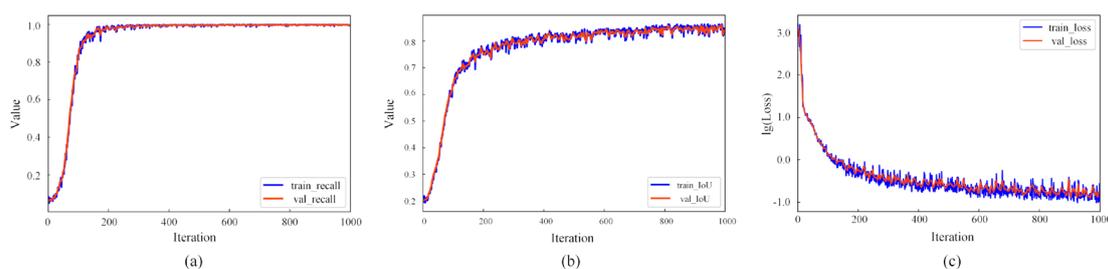
### 4.1. Performance of the Trained Excavator Detection Model

The excavator detection model was trained for approximately 200 epochs based on the training images according to the scheme in Section 3.1. In the training, the YOLOv3 parameters including the batch size, momentum, decay, and learning rate were initialized to 64, 0.9, 0.0005, and 0.0001, respectively, as shown in Table 2. These values are widely used by the scientific community [44,50] and we found that the subsequently trained excavator detection model had good accuracy. To avoid overfitting, extensive data augmentation approaches, including random crops, rotations, flips, hue, saturation, and exposure shifts, were used based on the training images. The training was conducted on a Linux workstation with an Intel i7 8700 CPU and a NVIDIA GeForce GTX 2080Ti graphics card.

**Table 2.** The initial values of the parameters in training the excavator detection model.

Batch Size	Momentum	Decay	Learning Rate
64	0.9	0.0005	0.0001

The three parameters (i.e., recall rate, IoU, and Loss) in the train set and validation set (see Section 3.1) for evaluating the trained model during the training process are shown in Figure 7 and Table 3. The validation set is verified every five iterations on the training set. Figure 7 clearly shows that the performance of the excavator detection model improves quickly after the beginning of the training. After 150 iterations (i.e., about 7 h), the recall rate with 0.5 as the objectness confidence threshold, IoU, and Loss on the validation set are 94.0%, 71.0%, and 0.637, respectively. After 500 iterations (i.e., about 20 h), the recall rate and IoU increase to 98.1% and 82.0%, respectively, and Loss decreases to 0.208, indicating that the model has good ability in detecting excavators from the validation set. After approximately 1000 iterations (i.e., about 40 h), the training stops with insignificant increases of the recall rate and IoU and a very slight decrease in loss. Then, the trained model was evaluated in the test set. In the test set, the trained model hardly omits the excavators and is able to detect the locations of the excavators accurately as shown in Table 3, which indicates that the trained model has a good generalization in detecting excavators. The recall rate and IoU values obtained in this study are higher than those reported in other studies [38,41] for multiple object detection training based on published datasets, such as the common objects in context [51] and ImageNet. The main reason is that the object to detect in this study is only assigned to one category and the excavator has an evident shape.



**Figure 7.** The recall rate, intersection over union (IoU), and loss of the train set and validation set for evaluating the trained model during the training process.

**Table 3.** Performance of the trained excavator detection model on the train set, validation set and test set.

Dataset	Recall Rate	Precision	Accuracy	IoU
train	99.8%	98.3%	98.0%	84.4%
validation	99.4%	98.0%	97.4%	83.7%
test	99.4%	97.7%	97.2%	83.1%

As described in Section 3.2.1 and shown in Figure 6, the image inputted into the excavator detection model is extracted from the video taken by the FLIR camera and resized simultaneously. Because the excavator detection model based on YOLOv3 is a fully convolutional network (see Section 3.1), it is able to take input images of different sizes and produce correspondingly sized feature maps without retraining the model [52]. The size of the inputted image affects the detection accuracy and processing speed, which directly determines the practicability of the UAV-ED system in real-world applications. Thus, the most appropriate size of the inputted image should be determined. The size of the image required by the YOLOv3 model should be a multiple of 32 and the default size of the published YOLOv3 model is  $416 \times 416$  pixels [43]. We found such a size can obtain a very good accuracy in the detection of the excavators; nevertheless, it can only process 3.7 images per second (namely, 0.27 s per image). Therefore, we further examined the processing speed and the corresponding detection accuracy for a series of sizes, ranging from  $416 \times 416$  pixels to  $288 \times 288$  pixels with a step of  $32 \times 32$ . This test was conducted on both the aforementioned Linux workstation as well as the Jetson TX2 embedded board. The details are shown in Table 4.

**Table 4.** The processing speed and detection accuracy of images with different resolutions on the computer and the Jetson TX2 embedded board.

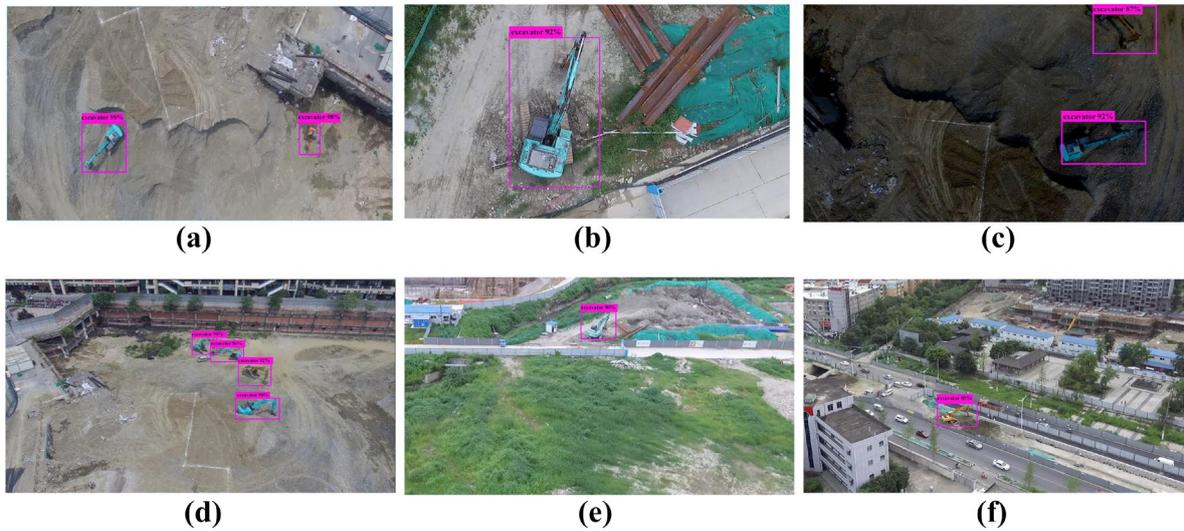
Resolutions	Processing Speed (FPS)		Detection Accuracy		
	Computer	Jetson TX2	Recall	IoU	Loss
$416 \times 416$	47	3.7	99.4%	83.7%	0.152
$384 \times 384$	50	4.3	98.3%	82.3%	0.155
$352 \times 352$	52	4.9	97.5%	81.0%	0.158
$320 \times 320$	58	5.4	96.8%	79.2%	0.165
$288 \times 288$	60	5.9	96.5%	78.9%	0.167

As expected, when the size of the inputted image decreases, the processing speed increases. With a size of  $288 \times 288$  pixels, the excavator detection model on the Jetson TX2 board can process 5.9 images per second (namely, 0.17 s per image) as shown in Table 4. Nevertheless, the detection accuracy only has slight variation:  $-2.9\%$  for the recall rate,  $-4.8\%$  for IoU, and 0.015 for Loss, suggesting that the detection accuracy can still well satisfy the excavator detection in real world applications. Therefore, the size of the image inputted into the excavator detection model was determined as  $288 \times 288$  pixels.

With a size of  $288 \times 288$  pixels, the trained excavator detection model was applied to the testing images. Examples of the detection results are shown in Figure 7a–c). Almost all the excavators were successfully detected, although the original images had different backgrounds, excavators of different colors and sizes, and varying illumination conditions. A closer look at Figure 7c demonstrates that a small yellow excavator in the upper right of the image has been detected, suggesting that the trained excavator detection model even has good applicability in the low illumination of early morning and evening and can distinguish the excavators from the background with a similar hue.

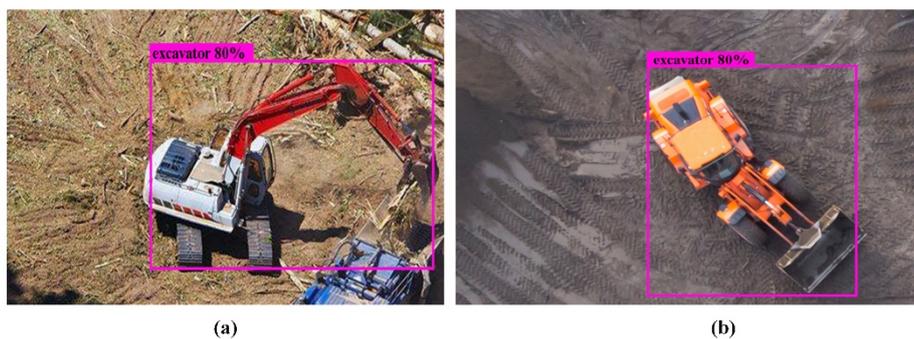
Users may require broader viewing extents with obliquely mounted cameras. Thus, the train excavator detection model was further applied to obliquely taken images, the spatial extents of which are larger than that of vertically taken images at the same flight height. Examples are shown in Figure 8d–f, which also have different backgrounds for the excavators. All the excavators were detected, indicating that the trained excavator detection model also has good performance for

obliquely taken images. This finding suggests that the UAV-ED system would be valuable for users that require broad viewing extents and less accuracy for the locations of the excavators.



**Figure 8.** The excavators detected from the testing images by the trained excavator detection model. The original images of (a), (b), and (c) were taken vertically by the UAV and that of (d), (e), and (f) were taken obliquely.

In the application of oil pipeline inspection, missed excavator detection may cause serious accidents. Therefore, the excavator detection model should detect all possible excavators, which may cause false alarms. Because the detection strategy of YOLOv3 (described in Section 3.1) is able to obtain negative samples from images containing excavators [43], images without excavators are not collected in our dataset (as mentioned in Section 2.1). In the test phase, it is found that not only excavators are detected by the trained excavator detection model, but similar objects are also detected. As shown in Figure 9, the grab machine and the crane in images, which do not harm the pipeline, may mistakenly be detected as excavators. This problem can be addressed by collecting more data, improving loss function [53] or the user's further judgement. When images containing suspicious targets are transmitted to users, users will further determine whether the detected targets are harmful to the pipeline.



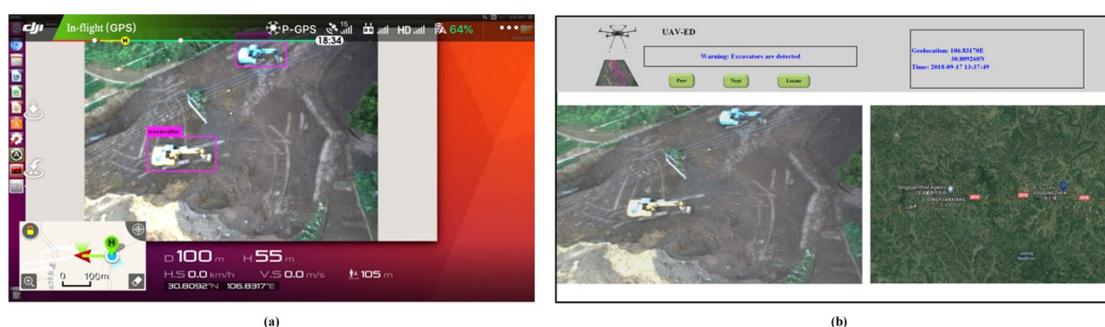
**Figure 9.** The images mistakenly detected as containing excavators. (a) Containing a grab machine; (b) containing a crane.

#### 4.2. Performance of the UAV-ED System

During the UAV remote-sensing experiment, several flights were conducted around the pipeline to evaluate the performance of the UAV-ED system. Before the experiment, a ground survey was conducted in the morning along the pipeline through walking to identify whether there were any excavators. We found there were four excavators working at two sites, approximately 10–

15 m away from the pipeline as shown in Figure 2. Since the excavators were working and moving, we recorded the geolocations of the two sites. The detection scope of the UAV-ED ranges from 40 m to 100 m on each side of the pipeline according to the flight height. Although the distance from the excavators detected by the UAV-ED to the pipeline may be beyond the lowest limitation (namely, 5 m; see Section 1), the staff of the Gas Transmission Management Division of PetroChina should be vigilant about this situation. The geolocation of the UAV is the approximation of the location of the detected excavator, but it is easy to find the detected excavator in the area specified by the UAV's geolocation because the excavator is noticeable.

To better test the performance of the UAV-ED system in the transmission of the detection results to the Warning Module, the computer with the Warning Module was placed in UESTC, which is approximately 350 km away from the testing area. During the flights, the real-time video processed by the UAV module could be displayed in the UAV Control Module as shown in Figure 10a. In addition, the detection results (i.e., the images of the detected excavators, the latitudes/longitudes, and time) were transmitted to the Warning Module as shown in Figure 10b.



**Figure 10.** Interfaces of the UAV Control Module and the Warning Module. (a) The UAV Control Module is able to display the real-time video processed by the UAV module; (b) the Warning Module is able to display the detection results (i.e., the images of the detected excavators, the latitudes/longitudes, and time) of the UAV-ED system.

The results showed that the UAV-ED system accurately detected all the excavators around the pipeline with different backgrounds and flight altitudes. Some examples are shown in Figure 11. The images displayed were acquired in different situations and accurate detection results were obtained from images with multiple excavators in Figure 11a, small excavators in Figure 11b, and those even containing only a portion of an excavator in Figure 11d. Furthermore, the flight altitudes in Figure 11e,f and the location of the excavator in Figure 11c did not affect the detection results. The detected geolocations of the excavators also agreed well with the geolocations of the sites recorded previously in the ground survey.



**Figure 11.** Images with detected excavators transmitted to the Warning Module during the UAV remote sensing experiment. The images were in different situations: (a) containing two excavators at Site A; (b) containing a small excavator at Site A; (c) containing an excavator in the corner of the image at Site A; and (d) containing a portion of an excavator at Site B. (e) and (f) containing the same excavators but acquired at different flight altitudes at Site B. All images were acquired at flight altitudes of 120 m, except (f), the altitudes of which was 60 m.

To test the time required to transmit the detection results to the Warning Module, two dedicated flights (i.e., flights I and II) were conducted at the two sites. The details of these two flights are shown in Table 5. The average consuming time from the UAV-ED system capturing an image to the Warning Module receiving the corresponding detection result is defined as the speed of the UAV-ED system.

**Table 5.** The number of the transmitted images and the corresponding durations of the two dedicated flights for testing the time required to transmit the detection results to the Warning Module.

Flight	Site	Number of Images	Duration (s)	Speed (s)
I	A	86	96 (10:47:17–10:48:53)	1.11
	B	70	72 (10:50:26–10:51:38)	1.02
II	A	148	173 (13:38:44–13:41:37)	1.16
	B	124	151 (13:45:32–13:48:03)	1.21
Overall		428	492 s	1.15

Flight I was in the morning. The UAV first hovered over Site A for 96 s; then, it flew to Site B and hovered there for 72 s. The Warning Module received 86 images with detected excavators as well as the corresponding geolocations for Site A and 70 images for Site B. The speeds were 1.11 s and 1.02 s, respectively. Flight II was in the afternoon. The UAV hovered over Site A for 173 s and Site B for 151 s. The speeds were 1.16 s and 1.21 s, respectively. For both of these flights, the 4G network was good and the overall speed was 1.15 s. Therefore, the transmission of the detection result based on the 4G network is fast enough for the user of the UAV-ED system to respond in real-time in areas with good 4G network.

## 5. Discussion

To the best of our knowledge, an evident difference of this study compared to other studies [8,12,22] is that the detection of excavators using UAV remote sensing is performed in real-time. The

approach proposed in this study realized the entire process for real-time object detection based on UAV remote sensing, including the excavator detection algorithm, the real-time processing of the remotely sensed data, and the real-time transmission of the detection results. Once the sensor onboard the UAV platform observes the excavator(s), the excavator detection model deployed on the embedded board can identify the excavator(s) in a very short time and the user can receive the related information in about 1.15 s. Such a fast response can strongly support timely decision-making. From this aspect, the approach developed in this study would outperform traditional approaches for processing UAV remote-sensing data. A possible alternative way for realizing the real-time processing is to transmit the UAV remote-sensing data to the cloud platform and then perform the processing. However, this way is limited by the transmission capability of the communication network due to the large size of the remote-sensing data. In contrast, the approach developed in this study has much a lower requirement on the transmission and thus, is faster and more economical.

Compared with the current method of excavator detection that is carried out in the field by laborers, the proposed approach is more economical. As shown in Table 6, the UAV inspection cost of 50 km of pipeline is approximately 141,000 yuan per year, which is less than half of that of manual inspection.

**Table 6.** The inspection cost of 50 km of oil pipeline per year using two different methods.

Inspection	Cost (×10,000 RMB)				Total
	Labor	Electricity	Communication	Equipment Depreciation	
Manual	28	0.1	0.5	0.4	29.0
UAV	4	0.5	0.1	9.5	14.1

In practical applications, the UAV inspection frequency can be changed to satisfy different needs. Users can perform multiple UAV inspections during a day in areas with high risk. In fact, this UAV inspection strategy is widely used in power line maintenance [54] and forest fire prevention [55].

The approach developed in this study can be extended easily to the detection of other objects in applications that requires the real-time processing of UAV remote-sensing data, such as the search and rescue of missing persons, the dynamic monitoring of power lines, the fighting of forest fires, and transportation management. Furthermore, it is reasonable to expect that the proposed real-time approach may be extended to the real-time estimation of surface parameters when the corresponding retrieval algorithms can be deployed on similar embedded boards and the UAV system can be adapted to other sensors (e.g., thermal infrared sensors, airborne-lidar, and multispectral sensors). The real-time estimation of surface parameters would be a benefit in fields such as precision agriculture, hydrology, and environmental science.

Nevertheless, there are some limitations in this study. First, the YOLOv3 model was selected in this study for excavator detection because it has low computational costs and is able to process images in real time on the embedded board. Although very high accuracy in detection has been obtained in this study, the accuracy may decrease for other target objects. In addition, due to the evident shape of the excavator, good accuracy can be obtained in the training of the excavator detection model based on only 350 images. For other complicated target objects, many more images may be required. Second, one should keep in mind that not all neural networks can be deployed on the embedded board used in this study and not every neural network deployed on the embedded board is able to process the UAV remote-sensing data in real time. Some neural networks are computationally heavy. Considering the computational ability of the current embedded board, it is difficult to process images in real time. Therefore, our future work will focus on deploying more neural networks, such as SegNet [56] and U-Net [57], on the embedded board to test their performance in practical applications. Third, the developed UAV-ED system is not able to realize continuous monitoring of the pipeline due to the short duration of the UAV flight. Nevertheless, the approach can be extended to ground cameras for continuous monitoring at fixed sites. Another issue

is that when the signal of 4G network is not good enough, the transmission of the warning messages may be slower. Nevertheless, areas with poor 4G networks usually have fewer human activities and the pipelines suffer fewer security threats. Thus, these areas can be inspected with fewer times by laborers.

## 6. Conclusions

Taking pipeline protection as an example, this paper proposed an approach to integrate UAV remote-sensing and deep-learning algorithms in order to realize the real-time detection of ground excavators. A system, termed UAV-ED, was further constructed for operational application. The UAV-ED system is composed of a UAV Control Module, a UAV Module, and a Warning Module. The UAV Control Module sets the flight missions of the UAV and controls its flight state. The UAV Module detects the excavator through the embedded board, determines the geolocation, and transfers the information to the other two modules. The Warning Module receives the real-time images of the excavator, its geolocation, and the time of acquisition.

The excavator detection model based on YOLOv3 is trained and deployed on the Jetson TX2 embedded board carried by the UAV. The excavator detection model reached 83.7% on average IoU and 99.4% for the recall rate with a threshold of 0.5 and 5.9 FPS for the detection speed on the Jetson TX2 embedded board. The excavator detection model and three other libraries, including the OpenCV library, the Libjpeg library, and the DJI Onboard SDK, are integrated with the UAV to form the UAV Module of the UAV-ED system. The UAV module is able to process the UAV remote-sensing data in real time. The processed UAV remote-sensing data, including the compressed JPEG image, the geolocation, and time are shown on the Warning Module of the UAV-ED system within about 1.15 s once the sensor mounted on the UAV module observes the excavator(s). The transmission between the UAV Module and the Warning Module is remote, wireless, and functions in real-time. The UAV Control Module of the UAV-ED system can display the video formed by the orderly extracted images with boxes on the detected excavators. The proposed approach and UAV-ED system were tested in different scenarios. The experimental results demonstrated that the UAV-ED system is able to detect excavators in real time and send the related results to users quickly. Thus, the UAV-ED system had good performance and would be a benefit for the management of pipeline safety.

The proposed approach can be easily extended to other applications that require real-time processing of UAV remote-sensing data. Further research will focus on three points to extend the application of UAV remote sensing: (i) the improvement of the network model for object detection with lower computational costs and higher accuracy; (ii) the implementation of the semantic segmentation network model on the Jetson TX2 embedded board, which is very useful in precision agriculture; and (iii) the labeling of the bounding box of the target with orientation, which is beneficial to reveal the direction of the object's motion and predict dynamic objects in videos [58].

**Author Contributions:** Conceptualization, L.M. and Z.P.; methodology, L.M. and J.Z. (Jirong Zhang); software, Z.L.; validation, L.M.; formal analysis, J.Z. (Jirong Zhang); investigation, Y.D.; resources, J.Z. (Jirong Zhang); data curation, L.M.; writing—original draft preparation, L.M.; writing—review and editing, J.Z. (Ji Zhou); visualization, J.Z. (Jirong Zhang); supervision, A.B.; project administration, J.Z. (Ji Zhou); funding acquisition, J.Z. (Ji Zhou). All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the Sichuan Province Science and Technology Plan Applied Basic Research Project (grant number: 2019YJ0205) and by the Chengdu Basic Innovation Research and Development Project (grant number: 2018-YF05-01475-GX) and by the Fundamental Research Funds for the Central Universities of China under grant ZYGX2019J069.

**Acknowledgments:** This work was supported in part by the Sichuan Province Science and Technology Plan Applied Basic Research Project (grant number: 2019YJ0205) and by the Chengdu Basic Innovation Research and Development Project (grant number: 2018-YF05-01475-GX) and by the Fundamental Research Funds for the Central Universities of China under grant ZYGX2019J069.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Ma, L.; Liu, Y.; Zhang, X.; Ye, Y.; Yin, G.; Johnson, B.A. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS J. Photogramm. Remote Sens.* **2019**, *152*, 166–177.
- Colomina, I.; Molina, P. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2014**, *92*, 79–97.
- Martínez-Carricondo, P.; Agüera-Vega, F.; Carvajal-Ramírez, F.; Mesas-Carrascosa, F.-J.; García-Ferrer, A.; Pérez-Porrás, F.-J. Assessment of UAV-photogrammetric mapping accuracy based on variation of ground control points. *Int. J. Appl. Earth Obs. Geoinf.* **2018**, *72*, 1–10.
- Wu, Z.; Ni, M.; Hu, Z.; Wang, J.; Li, Q.; Wu, G. Mapping invasive plant with UAV-derived 3D mesh model in mountain area—A case study in Shenzhen Coast, China. *Int. J. Appl. Earth Obs. Geoinf.* **2019**, *77*, 129–139.
- Bejiga, M.; Zeggada, A.; Nouffidj, A.; Melgani, F. A Convolutional Neural Network Approach for Assisting Avalanche Search and Rescue Operations with UAV Imagery. *Remote Sens.* **2017**, *9*, 100.
- Ballesteros, R.; Ortega, J.F.; Hernandez, D.; del Campo, A.; Moreno, M.A. Combined use of agro-climatic and very high-resolution remote sensing information for crop monitoring. *Int. J. Appl. Earth Obs. Geoinf.* **2018**, *72*, 66–75.
- Deng, L.; Mao, Z.; Li, X.; Hu, Z.; Duan, F.; Yan, Y. UAV-based multispectral remote sensing for precision agriculture: A comparison between different cameras. *ISPRS J. Photogramm. Remote Sens.* **2018**, *146*, 124–136.
- Madec, S.; Jin, X.; Lu, H.; De Solan, B.; Liu, S.; Duyme, F.; Heritier, E.; Baret, F. Ear density estimation from high resolution RGB imagery using deep learning technique. *Agric. For. Meteorol.* **2019**, *264*, 225–234.
- Sofonia, J.; Shendryk, Y.; Phinn, S.; Roelfsema, C.; Kendoul, F.; Skocaj, D. Monitoring sugarcane growth response to varying nitrogen application rates: A comparison of UAV SLAM LiDAR and photogrammetry. *Int. J. Appl. Earth Obs. Geoinf.* **2019**, *82*, 101878.
- Kim, I.-H.; Jeon, H.; Baek, S.-C.; Hong, W.-H.; Jung, H.-J. Application of Crack Identification Techniques for an Aging Concrete Bridge Inspection Using an Unmanned Aerial Vehicle. *Sensors* **2018**, *18*, 1881.
- Ammour, N.; Alhichri, H.; Bazi, Y.; Benjdira, B.; Alajlan, N.; Zuair, M. Deep Learning Approach for Car Detection in UAV Imagery. *Remote Sens.* **2017**, *9*, 312.
- Qu, T.; Zhang, Q.; Sun, S. Vehicle detection from high-resolution aerial images using spatial pyramid pooling-based deep convolutional neural networks. *Multimed. Tools Appl.* **2017**, *76*, 21651–21663.
- Zhu, J.; Sun, K.; Jia, S.; Lin, W.; Hou, X.; Liu, B.; Qiu, G. Bidirectional Long Short-Term Memory Network for Vehicle Behavior Recognition. *Remote Sens.* **2018**, *10*, 887.
- de Oliveira, D.; Wehrmeister, M. Using Deep Learning and Low-Cost RGB and Thermal Cameras to Detect Pedestrians in Aerial Images Captured by Multirotor UAV. *Sensors* **2018**, *18*, 2244.
- Zhang, X.; Chen, G.; Wang, W.; Wang, Q.; Dai, F. Object-Based Land-Cover Supervised Classification for Very-High-Resolution UAV Images Using Stacked Denoising Autoencoders. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 3373–3385.
- Morales, G.; Kemper, G.; Sevillano, G.; Arteaga, D.; Ortega, I.; Telles, J. Automatic Segmentation of *Mauritia flexuosa* in Unmanned Aerial Vehicle (UAV) Imagery Using Deep Learning. *Forests* **2018**, *9*, 736.
- Gevaert, C.M.; Persello, C.; Nex, F.; Vosselman, G. A deep learning approach to DTM extraction from imagery using rule-based training labels. *ISPRS J. Photogramm. Remote Sens.* **2018**, *142*, 106–123.
- Xiang, T.-Z.; Xia, G.-S.; Zhang, L. Mini-UAV-based Remote Sensing: Techniques, Applications and Prospectives. *arXiv* **2018**, arXiv:1812.07770.
- Zhao, Y.; Ma, J.; Li, X.; Zhang, J. Saliency Detection and Deep Learning-Based Wildfire Identification in UAV Imagery. *Sensors* **2018**, *18*, 712.
- Bah, M.; Hafiane, A.; Canals, R. Deep Learning with Unsupervised Data Labeling for Weed Detection in Line Crops in UAV Images. *Remote Sens.* **2018**, *10*, 1690.
- Kerkech, M.; Hafiane, A.; Canals, R. Deep leaning approach with colorimetric spaces and vegetation indices for vine diseases detection in UAV images. *Comput. Electron. Agric.* **2018**, *155*, 237–243.
- Kellenberger, B.; Marcos, D.; Tuia, D. Detecting mammals in UAV images: Best practices to address a substantially imbalanced dataset with deep learning. *Remote Sens. Environ.* **2018**, *216*, 139–153.
- Kim, D.; Liu, M.; Lee, S.; Kamat, V.R. Remote proximity monitoring between mobile construction resources using camera-mounted UAVs. *Autom. Constr.* **2019**, *99*, 168–182.
- Wikipedia. List of Pipeline Accidents. Available online: [https://en.wikipedia.org/wiki/List\\_of\\_pipeline\\_accidents](https://en.wikipedia.org/wiki/List_of_pipeline_accidents) (accessed on 10 December 2019).

25. EYETRAX. Security Cameras For The Oil And Gas Industries. Available online: <https://eyetrax.net/industries/oil-and-gas> (accessed on 10 December 2019).
26. Khoshboresh Masouleh, M.; Shah-Hosseini, R. Development and evaluation of a deep learning model for real-time ground vehicle semantic segmentation from UAV-based thermal infrared imagery. *ISPRS J. Photogramm. Remote Sens.* **2019**, *155*, 172–186.
27. Liu, X.; Yang, T.; Li, J. Real-Time Ground Vehicle Detection in Aerial Infrared Imagery Based on Convolutional Neural Network. *Electronics* **2018**, *7*, 78.
28. Loquercio, A.; Maqueda, A.I.; del-Blanco, C.R.; Scaramuzza, D. DroNet: Learning to Fly by Driving. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1088–1095.
29. Nex, F.; Duarte, D.; Steenbeek, A.; Kerle, N. Towards Real-Time Building Damage Mapping with Low-Cost UAV Solutions. *Remote Sens.* **2019**, *11*, 287.
30. Tijtgat, N.; Van Ranst, W.; Volckaert, B.; Goedeme, T.; De Turck, F. Embedded Real-Time Object Detection for a UAV Warning System. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (iccvw 2017), Venice, Italy, 22–29 October 2017; IEEE: New York, NY, USA, 2017; pp. 2110–2118, ISBN 978-1-5386-1034-3.
31. Hossain, S.; Lee, D. Deep-Learning Based Real-Time Multiple-Object Detection and Tracking from Aerial Imagery via Flying Robot with GPU-based Embedded Devices. *Sensors* **2019**, *15*, 3371.
32. Skydio Support. Skydio 2 Help. Available online: <https://support.skydio.com/hc/en-us> (accessed on 10 December 2019).
33. DJI. DJI. Available online: <https://www.dji.com/cn> (accessed on 10 December 2019).
34. The Little Ripper Life Sver. The Little Ripper. Available online: <https://thelittleripper.com.au> (accessed on 10 December 2019).
35. Github. Labellmg. Available online: <https://github.com/tzutalin/labellmg> (accessed on 10 December 2019).
36. 10086. Iot. Available online: <https://open.iot.10086.cn> (accessed on 10 December 2019).
37. Du, J.; Guo, J.; Xu, D.; Huang, Q. A Remote Monitoring System of Temperature and Humidity Based on OneNet Cloud Service Platform. In Proceedings of the 2017 IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS), Haining, China, 14–16 December 2017.
38. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv* **2013**, arXiv:1311.2524.
39. Girshick, R. Fast R-CNN. *arXiv* **2015**, arXiv:1504.08083.
40. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2015**, arXiv:1506.01497.
41. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2015**, arXiv:1506.02640.
42. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. *arXiv* **2016**, arXiv:1512.02325.
43. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
44. Zhang, X.; Yang, W.; Tang, X.; Liu, J. A Fast Learning Method for Accurate and Robust Lane Detection Using Two-Stage Feature Extraction with YOLO v3. *Sensors* **2018**, *18*, 4308.
45. Darket. YOLO. Available online: <https://pjreddie.com/darknet/yolo> (accessed on 10 December 2019).
46. Bradski, G. The OpenCV Library. *IEEE Robot. Autom. Mag.* **2009**, *16*, 100.
47. Opencv. Opencv. Available online: <https://opencv.org> (accessed on 10 December 2019).
48. JPEG. Libjpeg. Available online: <http://libjpeg.sourceforge.net> (accessed on 10 December 2019).
49. DJI. Developer. Available online: <https://developer.dji.com> (accessed on 10 December 2019).
50. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comput. Electron. Agric.* **2019**, *157*, 417–426.
51. COCO Dataset. Available online: <http://mscoco.org> (accessed on 10 December 2019).
52. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 8–10 June 2015.
53. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *arXiv* **2018**, arXiv:1708.02002.
54. Zhang, H.; Yang, W.; Yu, H.; Zhang, H.; Xia, G.-S. Detecting Power Lines in UAV Images with Convolutional Features and Structured Constraints. *Remote Sens.* **2019**, *11*, 1342.

55. Yuan, C.; Liu, Z.; Zhang, Y. Learning-Based Smoke Detection for Unmanned Aerial Vehicles Applied to Forest Fire Surveillance. *J. Intell. Robot. Syst.* **2019**, *93*, 337–349.
56. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *arXiv* **2015**, arXiv:1511.00561.
57. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv* **2015**, arXiv:1505.04597.
58. Xia, G.-S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A Large-scale Dataset for Object Detection in Aerial Images. *arXiv* **2017**, arXiv:1711.10398.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).