electronics

MDPI

Article

Real-Time Ground Vehicle Detection in Aerial Infrared Imagery Based on Convolutional Neural Network

Xiaofei Liu [1], Tao Yang [1,2,*] and Jing Li [3,*]

1 SAIIP, School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China; xiaofei@mail.nwpu.edu.cn
2 Research & Development Institute of Northwestern Polytechnical University in Shenzhen, Shenzhen 518057, China
3 School of Telecommunications Engineering, Xidian University, Xi'an 710071, China
* Correspondence: tyang@nwpu.edu.cn (T.Y.); jinglixd@mail.xidian.edu.cn (J.L.); Tel.: +86-150-0291-9079 (T.Y.); +86-139-9132-0168 (J.L.)

Received: 3 April 2018; Accepted: 19 May 2018; Published: 23 May 2018

Abstract: An infrared sensor is a commonly used imaging device. Unmanned aerial vehicles, the most promising moving platform, each play a vital role in their own field, respectively. However, the two devices are seldom combined in automatic ground vehicle detection tasks. Therefore, how to make full use of them—especially in ground vehicle detection based on aerial imagery–has aroused wide academic concern. However, due to the aerial imagery's low-resolution and the vehicle detection's complexity, how to extract remarkable features and handle pose variations, view changes as well as surrounding radiation remains a challenge. In fact, these typical abstract features extracted by convolutional neural networks are more recognizable than the engineering features, and those complex conditions involved can be learned and memorized before. In this paper, a novel approach towards ground vehicle detection in aerial infrared images based on a convolutional neural network is proposed. The UAV and the infrared sensor used in this application are firstly introduced. Then, a novel aerial moving platform is built and an aerial infrared vehicle dataset is unprecedentedly constructed. We publicly release this dataset (NPU_CS_UAV_IR_DATA), which can be used for the following research in this field. Next, an end-to-end convolutional neural network is built. With large amounts of recognized features being iteratively learned, a real-time ground vehicle model is constructed. It has the unique ability to detect both the stationary vehicles and moving vehicles in real urban environments. We evaluate the proposed algorithm on some low–resolution aerial infrared images. Experiments on the NPU_CS_UAV_IR_DATA dataset demonstrate that the proposed method is effective and efficient to recognize the ground vehicles. Moreover it can accomplish the task in real-time while achieving superior performances in leak and false alarm ratio.

Keywords: aerial infrared imagery; real-time ground vehicle detection ; convolutional neural network; unmanned aerial vehicle

1. Introduction

Vehicle detection is an essential and pivotal role in several applications like intelligent video surveillance [1–4], car crash analysis [5], autonomous vehicle driving [6]. Most traditional approaches adopt the way that the camera is installed on a low-altitude pole or mounted on the vehicle itself. For instance, Sun and Zehang [7] present a method which jointly uses Gabor filters and Support Vector Machines for on-road vehicle detection. The Gabor filters are for feature extraction and these extracted features are used to train a classifier for detection. The authors in [8] propose a method to detect

vehicles from stationary images using colors and edges. Zhou, Jie and Gao [9] propose a moving vehicle detection method based on example-learning. With regard to these approaches, the coverage of camera is limited despite rotating in multiple directions, they only detect vehicles on a small scale.

Given the installed camera's limited coverage, researchers turn to other moving platforms. Satellites [10,11], aircrafts, helicopters and unmanned aerial vehicles have been used to solve the bottleneck. The cost of images collected by the satellites, aircrafts and helicopters is remarkably high, and this equipment isn't able to make a quick response according to the time and weather. With the rapid development of the unmanned aerial vehicles industry, the price of small drones has dropped in recent years. The UAVs(unmanned aerial vehicles) have been the research focus. It seems easily accessible for the general public to obtain it, so all kinds of cameras including optical and infrared, have started to be installed on it. In this way, the unmanned aerial vehicle can be used as a height–adjustable moving camera platform on a large scale. Many researchers have made great efforts in this field. For example, Luo and Liu [12] propose an efficient static vehicle detection framework on aerial range data provided by the unmanned aerial vehicle, which is composed of three modules–moving vehicle detection, road area detection and post processing. The authors in [13] put forward a vehicle detection method from UAVs, which is integrated with Scalar Invariant Feature Transform and Implicit Shape Model. Another work [14], a hybrid vehicle detection method that integrates the Viola–Jones (V–J) and linear SVM classifier with HOG(Histogram of Oriented Gradient) features, is proposed for vehicle detection for aerial vehicle images obtained in low-altitude. It is not able to choose a robust feature to aim at the small size vehicles. Besides, some researchers in [15–17] have adopted the other sensors (e.g., depth sensors, RGB-D imagery) into the object detection area.

Another vehicle detection methods are accomplished by background modeling or foreground segmentation. The authors in [18] put forward with a method of moving object detection on non–stationary cameras and bring it to vehicle detection on mobile device. They model the background through dual-mode single Gaussian Model with life–cycle model and compensate the motion of the camera via mixing neighboring models. The authors in [19,20] propose a method of detecting and locating moving object under realistic condition based on the motion history images representation, which incorporates the timed–MHI for motion trajectory representation. Afterwards, a spatio–temporal segmentation procedure is employed to label motion regions by estimating density gradient. However these methods might cause a great number false alarms and fail to detect the stationary vehicles.

All these adaptive detection methods above are same in essentials while differing in minor points. They employ the similar strategy: manual designed features (e.g., SIFT, SURF, HOG, Edge, Color or their combinations) [21–23], background modeling or foreground segmentation, common classifiers (e.g., SVM, Adaboost) and sliding window search. These manual features might not hold the diversity of vehicles' shapes, illumination variations and background changes. The sliding window is an exhaustive traversal, which is time–consuming, not purposeful. This might cause too many redundant bounding boxes and has a bad influence on the following extraction and classification's speed and efficiency.

However, deep learning [24–29] establishes convolutional neural network that could automatically extract abundant representative features from the vast training samples. It has an outstanding performance on diverse data. Lars et al. [25] propose a network for vehicle detection in aerial images, which has overcome the shortcoming of original approach in case of handling small instances. Deng et al. [26] propose a fast and accurate vehicle detection framework, they develop an accurate–vehicle– proposal–network based on hyper feature map and put forward with a coupled R-CNN(convolutional neural network) method. A novel double focal loss convolutional neural network is proposed in [27]. In this paper, the skip connection is used in the CNN structure to enhance the feature learning and the focal loss function is used to substitute for conventional cross entropy loss function in both the region proposed network and the final classifier. They [27,30,31] all adopt the same framework, Region Proposal plus Convolutional Neural Network. By virtue of the CNN's strong feature extraction capacity, it achieves a higher detection ratio. Inspired by these work above,

the authors in [29,32] introduce the elementary framework on aerial vehicle detection and recognition. As described in [29], a deep convolutional neural network is adopted to mine highly descriptive features from candidate bounding boxes, then a linear support vector machine is employed to classify the region into "car" or "no–car" labels. The authors in [32] propose a hyper region proposal network to extract potential vehicles with a combination of hierarchical feature maps, then a cascade of boosted classifiers are employed to verify the candidate regions, false alarm ratio is further reduced.

All these work above have achieved tremendous advances in vehicle detection [33,34]. For object detection, images matching plays a vital role in searching part. The authors in [33] propose a novel visible-infrared image matching algorithm, and they construct a co–occuring feature by cross-domain image database and feature extraction. Jing et al. [34] extend the visible–infrared matching to photo-to-sketch matching by constructing visual vocabulary translator. The authors in [15] extract object silhouettes from the noisy background by a sequence of depth maps captured by a RGB-D sensor and track it using temporal motion information from each frame. The authors in [17] present a novel framework of 3D object detection, tracking and recognition from depth video sequences using spatiotemporal features and modified HMM. They use spatial depth shape features and temporal joints features to improve object classification performance. However, those approaches are not suitable for aerial infrared vehicle detection. The vehicle detections based on deep neural network and classification can't reach the real-time demands. The vehicle detections based on these manual designed features' matching have poor performances on the detection ratio measurement, for the aerial infrared images are low resolution and fuzzy and the manually extracted features are rare.

Considering the trade-off between the real-time demand and quantified index–*Precision*, *Recall* and *F*1-*score*, we adopt the convolutional neural network (the number of layers is not deep) to extract abundant features in the aerial infrared images, treat the vehicle detection as a typical regressive problem to accelerate the bounding boxes generations. Some detection results are illustrated in Figure 1. The majority of vehicles are detected, and these bounding boxes approximately cover the vehicles. The proposed method unexpectedly runs at a sampling frequency of 10 fps. The real-time vehicle detection is demanding. In the detection system, we might not demand an extremely accurate vehicle position, but an approximate position obtained in time is more necessary. Once detection speed falls behind the sample frequency, the information provided is lagged. This might mislead surveillance system.

The main contributions of this paper can be summarized as follows:

- We propose a method of detecting ground vehicles in aerial imagery based on convolutional neural network. Firstly, we combine the UAV and infrared sensor to the real-time system. There exist some great challenges like scale, view changes and scene's complexity in ground vehicle detection. In addition, the aerial imagery is always low-resolution, fuzzy and low-contrast, which adds difficulties to this problem. However, the proposed method adopts a convolutional neural network instead of traditional feature extraction, and uses the more recognized abstract features to search the vehicle, which have the unique ability to detect both the stationary and moving vehicles. It can work in real urban environments at 10 fps, which has a better real-time performance. Compared to the mainstream background model methods, it gets double performances in the *Precision* and *Recall* index.
- We construct a real-time ground vehicle detection system in aerial imagery, which includes the DJI M-100 UAV (Shenzhen, China), the FLIR TAU2 infrared camera (Beijing, China.), the remote controls and iPad (Apple, California, US). The system is built to collect large amounts of training samples and test images. These images are captured on different scenes includes road and multi-scenes. Additionally, this dataset is more complex and diversified in vehicle number, shape and surroundings. The aerial infrared vehicle dataset (The dataset (NPU_CS_UAV_IR_DATA) is online at [35],) which is convenient for the future research in this field.

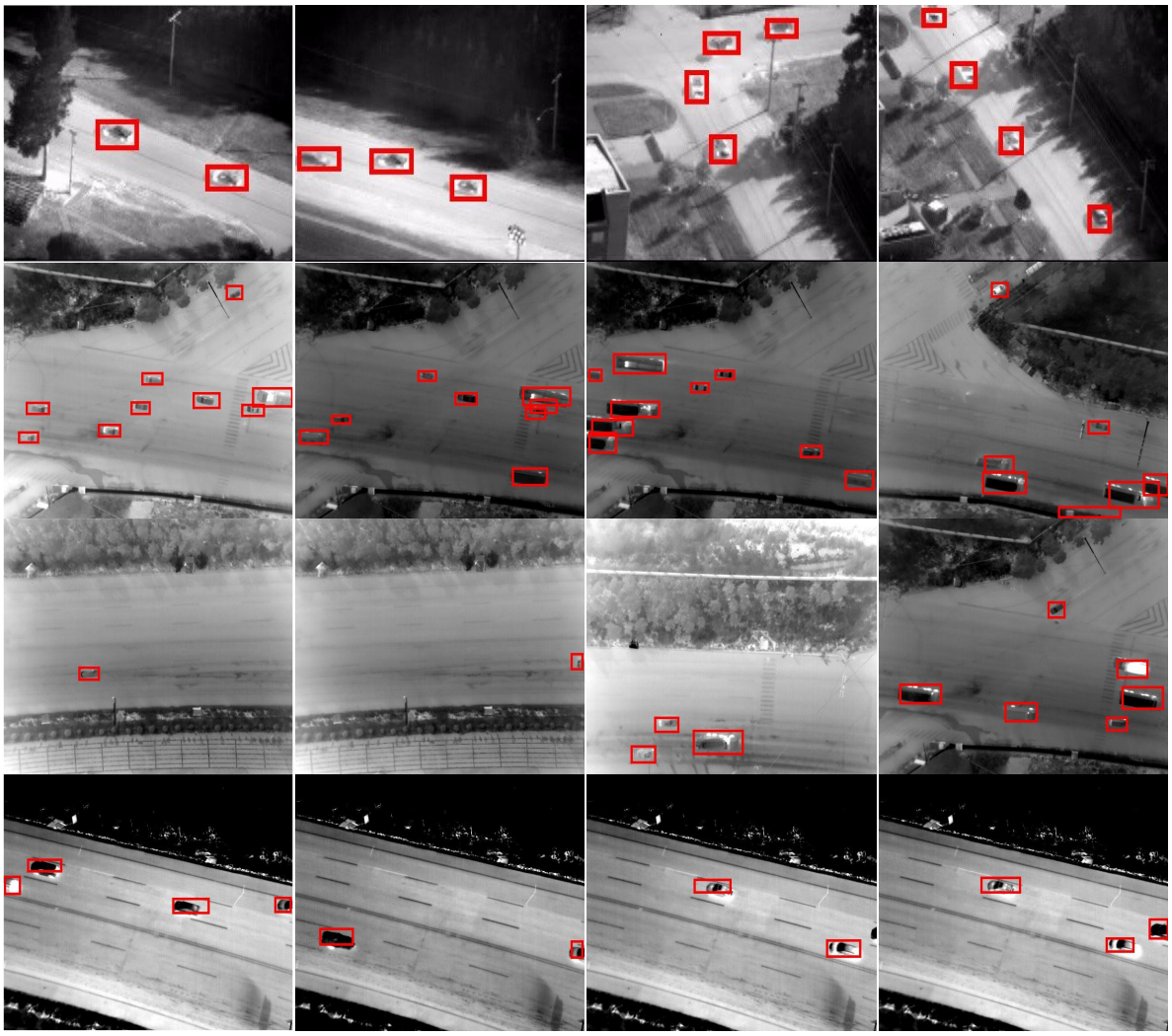

Figure 1. The detection method's performances on four tests, from top to bottom: VIVID_pktest1 [36], NPU_DJM100_1, NPU_DJM100_2, Scenes Change [35].

2. Aerial Infrared Ground Vehicle Detection

The proposed method is illustrated in Figure 2. It can be mainly divided into three steps. First, we manually segment vehicles by the help of a *labelimg* toolbox [37]. The labeled results are shown in Figure 3. This labeling step is pivotal to training [38]. The second step is devoted to sample region feature extraction in a convolutional neural network. We use data augmentations like rotation, crops, exposure shifts and more to expand samples. For training, we adopt a pre-trained classification network on ImageNet [39], and then fine-tune this. The pre-trained model on the ImageNet has many optimization parameters. On the basis of this, the loss function can be convergent rapidly in the training process. We add a region proposal layer to predict vehicles' coordinates (x, y, width, height) and corresponding confidence. These outputs contain many false alarms and redundant bounding boxes. We remove false alarms by confidence threshold. Finally, non-maximum suppression is adopted to eliminate redundant bounding boxes.

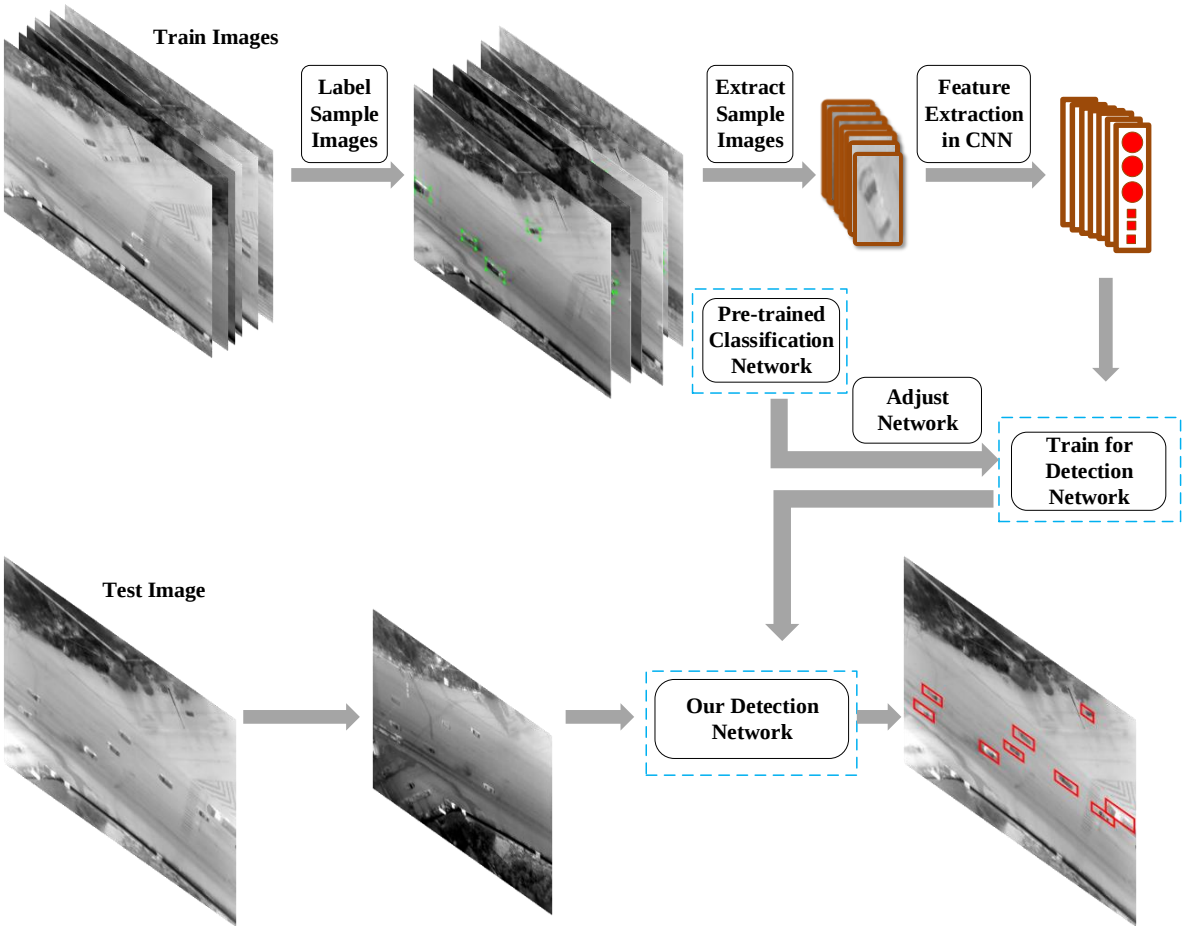

Figure 2. Flowchart of the proposed vehicle detection method in infrared images. Before training, we manually label vehicles in images via *labelimg* tool. In the label, we rectangle the vehicle by locating the top left corner and down right corner, and then keep these location and label information as a xml file. Afterwards, it is necessary to expand the sample by the operations—rotation, crop and shift. We load a pre-trained classification network for training. The Pre-Trained Network: classification network pre-trained on ImageNet [39].

2.1. Label Train Samples

Before labeling, it is necessary to construct an aerial infrared system to capture images for training samples. The aerial infrared system is mainly composed of the DJI Matrice 100 and the FLIR TAU2 camera. The DJI Matrice 100's major components are made of carbon fiber, which makes it light and solid, in order to guarantee it flies smoothly. The infrared sensor possesses the ability of temperature measurement and various color models' conversion, which meets the rigorous demands in several environments. In the capture, an intersection filled with a large volume of traffic is chosen as a flight place. Aerial images are captured at five different times alone. Images chosen from the first four times are train samples. Furthermore, aerial infrared vehicle samples from the public data VIVID_pktest are added.

Before training, it is necessary to label large amounts of training samples. These green rectangular regions rectangled in Figure 3 are some labeled samples. Partial vehicles appear in the image due to the limited view of infrared sensors, especially when it turns a corner, passes through the road or starts to enter into view, so we may catch the front or rear of some vehicles. These pieces of information are helpful because the vehicles often pass through an intersection or make a turn. This information mentioned above can ensure sample integrity. The information captured is crucial for vehicle detection.

In the label process, we obtain some vehicle patches in the infrared images. This guarantees that more training samples are captured and more situations are collected as much as possible. Although this operation is time-consuming and implemented offline, it insures vehicle samples' integrity [38]. This can avoid rough sample segmentation. We could observe some part vehicles in the left in 1–3 (row-col) in Figure 3. This is because the vehicle starts to come into view. There are some moving vehicles close to each other in (3–4) and (2–3). Once roughly segmented, the neighborhoods could be mistaken for just one, but there are two or more vehicles in practice.

All the vehicles are labeled in the training sample, then each image and vehicle position are made into a xml format as the voc [40].

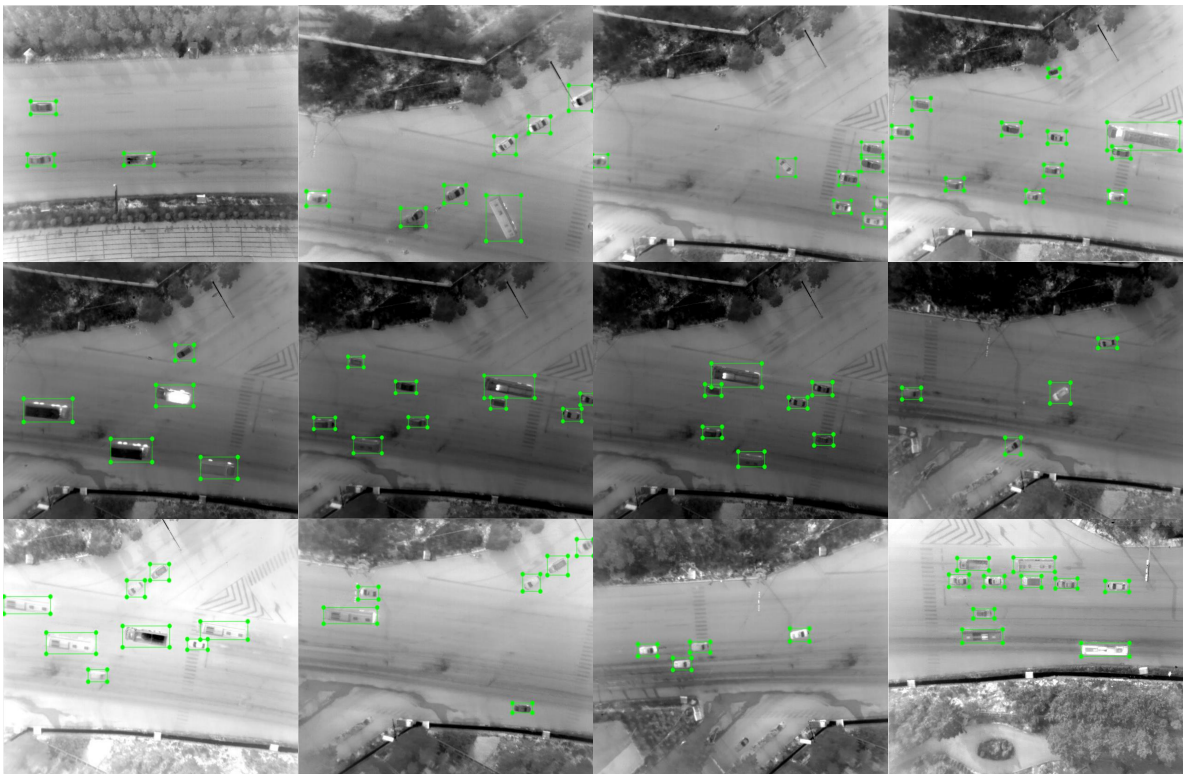

Figure 3. The labeled vehicle samples captured by the unmanned aerial vehicle DJI MATRICE-100. The manual label process is accomplished by the *labelimg* toolbox, which is a widely used tool in the sample label. Firstly, the label "vehicle" is written in this tool before label process. Then, we put green rectangles around the vehicles in images by searching the two locations: the left-top corner and the right-bottom corner. Finally, we keep this position and label information in a xml format like the voc.

2.2. Convolutional Neural Network

With respect to vehicle detection in aerial infrared images, we apply a convolutional neural network to the full image. It is based on the regressive idea to accomplish the object (vehicles) detection, rather than a typical classification problem. The network designed extracts features and trains on the full images, not the local positive and negative samples. The neural network's architecture is shown in Figure 4. Firstly, we resize the input image into 416×416, and utilize the convolutional layer and pooling layer by turning to an extract feature. Inspired by the fact that the Faster R-CNN [31] predicts offset and confidence for bounding boxes using the region proposal, we adopt a region proposal layer to predict bounding boxes. A lot of bounding boxes are obtained this way. We remove some false bounding boxes with low confidence by a threshold filter, and then eliminate redundant boxes using the non maximum suppression.

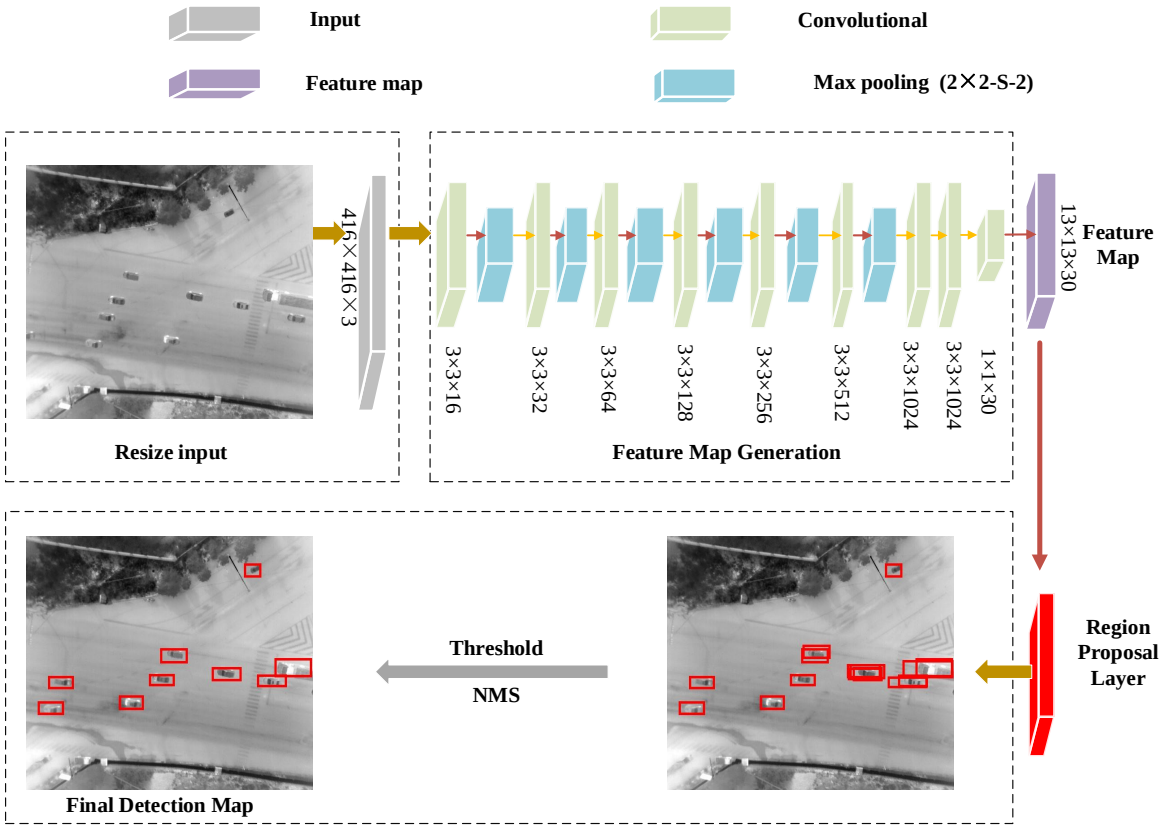

Figure 4. The architecture: The detection network is composed of nine convolutional layers, six pooling layers and a region proposal layer. Input image : 416 × 416 × 3. Output feature map: 13 × 13 × 30. The convolutional layers: 3 × 3 filters; The pooling layers: max pooling (2 × 2 with 2 stride). After each pool, filter channels double. We add a region proposal layer following the feature map, which is designed to generate bounding boxes, and then carry out threshold and NMS (Non Maximum Suppression) disposal.

Feature Map Generation

The detection framework can be mainly divided into two parts: **feature map generation** and **candidate bounding boxes generation**. The details of feature map are illustrated in Table 1. The process is composed of 15 layers: nine convolutional layers and six max pooling layers. Table 1 illustrates the filters channel, size, input and output of each layer. The original image is resized into 416 × 416 as the input. The convolutional layers downsample it by a factor 32, and the output size is 13 × 13. After this, there exists a single center cell in the feature map. The location prediction is based on the center location mechanism.

We carry out 16 filters (3 × 3) convolution operation on the input (416 × 416 × 3), followed by a 2 × 2 with two strides. Subsequently, the number of filers doubles, but the number of strides for pooling layer remains unchanged. Executing the above operations until the number of filters increases to 512, then the channel of stride on pooling layer is set as 1. This disposal wouldn't change the channel of the input (13 × 13 × 512). Based on this, we add two 3 × 3 convolutional layers with 1024 filters, following a 1 × 1 convolutional layer with 30 filters. Finally, the original image is turned into 13 × 13 × 30.

Table 1. Feature Map Generation includes the input and output of each layer.

Number	Layer	Filters	Size/Stride	Input	Output
0	convolutional	16	3×3/1	416 × 416 × 3	416 × 416 × 16
1	max pooling		2 × 2/2	416 × 416 × 16	208 × 208 × 16
2	convolutional	32	3 × 3/1	208 × 208 × 16	208 × 208 × 32
3	max pooling		2 × 2/2	208 × 208 × 32	104 × 104 × 32
4	convolutional	64	3 × 3/1	104 × 104 × 32	104 × 104 × 64
5	max pooling		2 × 2/2	104 × 104 × 64	52 × 52 × 64
6	convolutional	128	3 × 3/1	52 × 52 × 64	52 × 52 × 128
7	max pooling		2 × 2/2	52 × 52 × 128	26 × 26 × 128
8	convolutional	256	3 × 3/1	26 × 26 × 128	26 × 26 × 256
9	max pooling		2 × 2/2	26 × 26 × 256	13 × 13 × 256
10	convolutional	512	3 × 3/1	13 × 13 × 256	13 × 13 × 512
11	max pooling		2 × 2/1	13 × 13 × 512	13 × 13 × 512
12	convolutional	1024	3 × 3/1	13 × 13 × 512	13 × 13 × 1024
13	convolutional	1024	3 × 3/1	13 × 13 × 1024	13 × 13 × 1024
14	convolutional	30	1 × 1/1	13 × 13 × 1024	13 × 13 × 30

2.3. Bounding Boxes Generation

After convolutional and pooling operations, the final output is a 13 × 13 × 30 feature map. We add a region proposal layer following the feature map to predict the vehicle's location. Inspired by the RPN (region proposal network) of Faster-RCNN [31], we adopt a region proposal layer to service for vehicle border regression. The core purpose of the region proposal layer is to directly generate region proposals by the convolutional neural network. To generate region proposals, we slide a small network over the feature map output by the last shared convolutional layer. The small network takes a 3 × 3 spatial window as input on the feature map. The sliding window is mapped to a 30-dimensional feature vector. The feature is fed into a box-regression layer this way.

At each sliding-window, we simultaneously obtain a great deal of region proposals. Supposing the number of the proposals for each location is R, the output of region proposal layer is $4R$ coordinates, which are the R boxes' parametric expressions. The five classes about the proposals' percentages of width and height are (1.08, 1.09), (3.42, 4.41) , (6.63, 11.38) , (9.42, 5.11), (16.62, 10.52).

2.3.1. Vehicle Prediction on Bounding Boxes

The detection network is an end-to-end neural network. The vehicle's bounding boxes are accomplished directly by the network, The bounding boxes are achieved in the **bounding boxes generation** section. The confidence is computed as Equation (1):

$$C = P_{vehicle} * I_{pred}^{truth}, \tag{1}$$

where $P_{vehicle}$ indicates whether there exists a vehicle in the current prediction box, the I_{pred}^{truth} is the intersection over union between the predicted box and the ground truth. If no vehicle, the $P_{vehicle}$ is 0, 1, otherwise.

The confidence reflects the confidence level if the box contains a vehicle. A new parameter: $confidence$ is added, and each bounding box can be parameterized by $x, y, w, h, confidence$.

During the practical evaluating process, these above values are normalized to the range of $[0, 1]$. The $confidence$ reflects the probability of predicted boxes belonging to the vehicle. The $P_{vehicle}$ is defined as follows:

$$P_{vehicle} = \begin{cases} 1, & \text{vehicle,} \\ 0, & \text{no vehicle.} \end{cases} \tag{2}$$

2.3.2. Non Maximum Suppression

In order to eliminate redundant bounding boxes, we use non maximum suppression to find the best bounding box for each object. It is used to suppress non-maxima elements and search the local maxima value. The NMS [41,42] is for selecting high score detections and skipping windows covered by a previously selected detection.

The left-top corner (X_{min},Y_{min}), right-bottom corner (X_{max},Y_{max}), and $confidence$ of detection boxes are the inputs in the NMS. The (X_{min},Y_{min}) and (X_{max},Y_{max}) are calculated by the following equations:

$$X_{min} = x - w, \tag{3}$$
$$X_{max} = x + w, \tag{4}$$
$$Y_{min} = y - h, \tag{5}$$
$$Y_{max} = y + h. \tag{6}$$

The area of each bounding box is calculated by Equation (7):

$$area = (X_{max} - X_{min} + 1) * (Y_{max} - Y_{min} + 1). \tag{7}$$

Then, the bounding boxes are sorted by confidence, and the overlap area of box i and j is computed by Equation (12):

$$X_{cross1} = max(X_{min}(i), X_{min}(j)), \tag{8}$$
$$Y_{cross1} = max(Y_{min}(i), Y_{min}(j)), \tag{9}$$
$$X_{cross2} = min(X_{max}(i), X_{max}(j)), \tag{10}$$
$$Y_{cross2} = min(Y_{max}(i), Y_{max}(j)), \tag{11}$$

$$cover_{(i,j)} = \frac{(X_{cross2} - X_{cross1} + 1) * (Y_{cross2} - Y_{cross1} + 1)}{min(area(i), area(j))}. \tag{12}$$

Once the $cover_{(i,j)}$ is over the suppression threshold, the bounding box with lower confidence would be discarded and the bounding box with highest confidence would be finally kept.

We unconditionally retain the box with higher confidence in each iteration, then calculate the overlap percentage between the box with the highest confidence and the other boxes. If the overlap percentage is bigger than 0.3, the current iteration terminates. The best box is determined until all the regions have been traversed.

3. Aerial Infrared System and Dataset

How we obtain the aerial infrared images (equipments and flight height) and prepare training samples and test images will be demonstrated in this section. In the test, we verify the method on the **VIVID_pktest1** [36], which has a pretty outstanding performance. However, these images in VIVID_pktest1 can not represent the aerial infrared images in the actual flight.

We capture the actual aerial infrared images (five different times) at an intersection. Experiments are implemented based on the Darknet [43] framework and run on a graphics mobile workstation with Intel core i7-3770 CPU (Santa Clara, California, US), a Quard K1100M of 2 GB video memory, and 8 GB of memory. The operating system is Ubuntu 14.04 (Canonical company, London, UK).

3.1. Aerial Infrared System

To evaluate the proposed vehicle detection approach, we have constructed an aerial infrared system, which is composed of the DJI Matrice 100 and the FLIR TAU2 camera.

Experiments are conducted by using aerial infrared images with 640 × 512 resolution, which are captured by a camera mounted on a quad rotor with a flight altitude of about 120 m above the ground. Figure 5 shows the basic components of the system, its referenced parameters are illustrated in Table 2. The dataset is online at [35].

Table 2. The equipment and parameters.

Camera and UAV	Specification	Parameter
	aircraft	DJI-MATRICE 100
	infrared sensor	FLIR TAU2
	capture solution	640 × 512
	capture frame rate	10 fps
	focal length	19 mm
	head rotation	32° × 26°

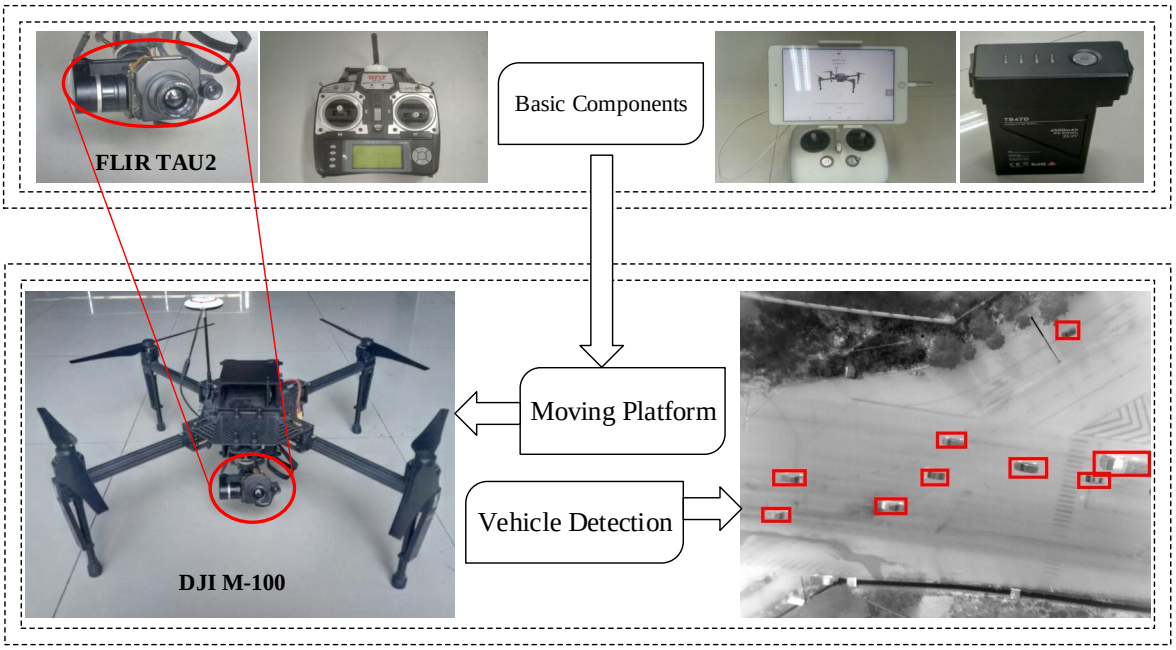

Figure 5. The real-time detection system is mainly composed of the DJI M-100, the FLIR TAU2 camera, the color model remote control, the flight remote control unit and iPad. The sensor installed on the UAV can capture the ground vehicles in real time, then transmit this information to the processor, and finally the processor shows the real-time detector on the screen.

3.2. Dataset

3.2.1. Training Samples

VIVID_pktest Sample: As we all know, the VIVID is a public data set for object tracking, which is composed of three subparts. The second part(VIVID_pktest2) [44] is a training sample. The image sequences are continuous in time, and the adjacent frames are similar to each other. If all images are put into training, the samples are filled with redundancy, so we only choose a set of 151, but which cover all vehicles appearing in VIVID_pktest2.

The authentical infrared sample: For the actual aerial infrared images, an intersection filled with large traffic volume is chosen as a flight place. We capture vehicle samples at five different times alone and choose sample images from the the first four times. The sampling frequency is 10 fps. Finally, we select 368 images, each of which is different in vehicle number, shape and color, as training samples considering redundant samples.

3.2.2. Test Images

As for evaluating the proposed method, we prepare four aerial infrared test image groups. The NPU_DJM100_1, NPU_DJM100_2 and Scenes Change are all captured over Xi'an, China. Four scenes with different backgrounds, flying altitudes, recording times and outside temperatures are used for testing (seen Table 3).

Table 3. Basic information of four test image groups.

Test	Size	Flying Altitude	Scenario	Date/Time	Temperature
VIVID_pktest1	320 × 240	80 m	Multi-scene	Unknown	Unknown
NPU_DJM100_1	640 × 512	120 m	Road	18 May 2017/16:00 pm	29° C
NPU_DJM100_1	640 × 512	120 m	Road	18 May 2017/16:30 pm	29° C
Scenes Change	640 × 512	80 m	Road	14 April 2017/10:30 pm	18° C

- **VIVID_pktest1**: The VIVID_pktest1 [36] is the first test image group, which is used for testing the detection network trained by the sample from the VIVID_pktest2 [44]. The VIVID_pktest1 is captured at an 80 m high altitude, which contains 100 images and 446 vehicles. The size is 320 × 240.
- **NPU_DJM100_1**: The sample chosen and their adjacent images from the aerial infrared images captured in the first four times is removed, then the remaining is used as the second test image group.
- **NPU_DJM100_2**: The images captured at the fifth time are the third test image group. There are few connections with images belonging to the previous four times.
- **Scenes Change**: The images are captured at earlier times and 80 m flight height. There is not a lot of traffic. This scene is totally different from all of the above. It is used to eliminate scenario training possibilities.

3.3. Training

In training, we use a batch size of 32, a max batch of 5000, a momentum of 0.9 and a decay of 0.0005. Through the training, the learning rate is set as 0.01. In each convolutional layer, we implement a batch normalized disposal except for the final layer before the feature map. With respect to the exquisitely prepared sample images, we divide them at a ratio of 7:3. Seventy percent were used for training, the remaining is for validation.

Loss function: In the objective module, we use the Mean Squared Error (MSE) for training.

$$Loss = \sum_{i=0}^{S^2} coordError + iouError + classError, \tag{13}$$

where S is the dimension's number of the network's output, $coordError$ is the error of coordinates between the predicted and the labeled, $iouError$ is the overlap's error, and $classError$ is the category of error (vehicle or non-vehicle). In the experiment, we amend Equation (13) by the following:

(1) The coordinates and the IOU (intersection over union) have different contribution degrees to the Loss, so we set the $\lambda_{coord} = 5$ to amend the $coordError$.

(2) For the IOU's error, the gridding includes the vehicle and the gridding having no vehicle should make various contributions to $Loss$. We use the $\lambda_{noobj} = 0.5$ to amend the $iouError$.

(3) As for the equal error, these large objects' impacts should be lower than small ones on vehicle detection because the percentage of error belonging to large objects is far less than those belonging to small ones. We square the w, h to improve it. The final Loss is as Equation (14):

$$Loss = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \prod_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \prod_{ij}^{obj} [(C_i - \hat{C}_i)^2] + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \prod_{ij}^{noobj} [(C_i - \hat{C}_i)^2]$$
$$+ \sum_{i=0}^{S^2} \prod_{ij}^{obj} \sum_{c \in classes} [(p_i(c) - \hat{p}_i(c))^2], \quad (14)$$

where the x,y,w,h,C,p are the predicted, and the $\hat{x}$, $\hat{y}$,$\hat{w}$,$\hat{h}$,$\hat{C}$,$\hat{p}$ are the labeled. The $\prod_{ij}^{obj}$ and the $\prod_{ij}^{noobj}$ reflect that the probability of that object is in, and not in, the j bounding boxes, respectively.

4. Experimental Results and Discussion

The method's performances on four test image groups are respectively shown in Figures 6–9. We rectangle the vehicles with red color. Some representative detection results will be demonstrated in **achievement exposition** section. The concert efficiency is given in the **statistical information** section.

4.1. Achievement Exposition

As seen in Figure 6, almost all of the vehicles have been detected by the proposed method. There exist many shadows of trees in the 2-3 (row-col) of Figure 6. This would cause great disturbances to vehicle detection tasks. This problem can be solved by storing and learning the similar cases before. In (2-2) of Figure 6, when the vehicle abruptly turns in the intersection, it might escape from surveillance. However, the method could catch the tendency and locate it in time. The test images are of good quality in the VIVID_pktest1 [36] and they haven't yet involved more complicated conditions like large illuminations. The test images of VIVID_pktest1 [36] are much simpler than the real aerial images in both the number of vehicles and the conditions' complexity. The performances of the VIVID_pktest1 [36] were not sufficiently convincing, hence the actual aerial infrared images obtained by the DJI Matrice 100 are used to test the method.

Figure 6. The performances of the method on part of the images of **VIVID_pktest1** [36].

As Figure 7 shows, the NPU_DJM100_1 is more challenging in vehicle's quantity and environmental complexity. There exist distinct illumination variations in the (1-1) and (2-1) of Figure 7. The traffic flow is very large in the intersection and the vehicles shuttling back and forth is very common. There are two detection boxes on the same vehicle in (2-2). There are two rectangles put around the same vehicle

on the right but little overlap. The suppression threshold can not be set as a very small value, as this would have a bad influence on eliminating redundant rectangles. There still exist several false alarms in (2-4) of Figure 7.

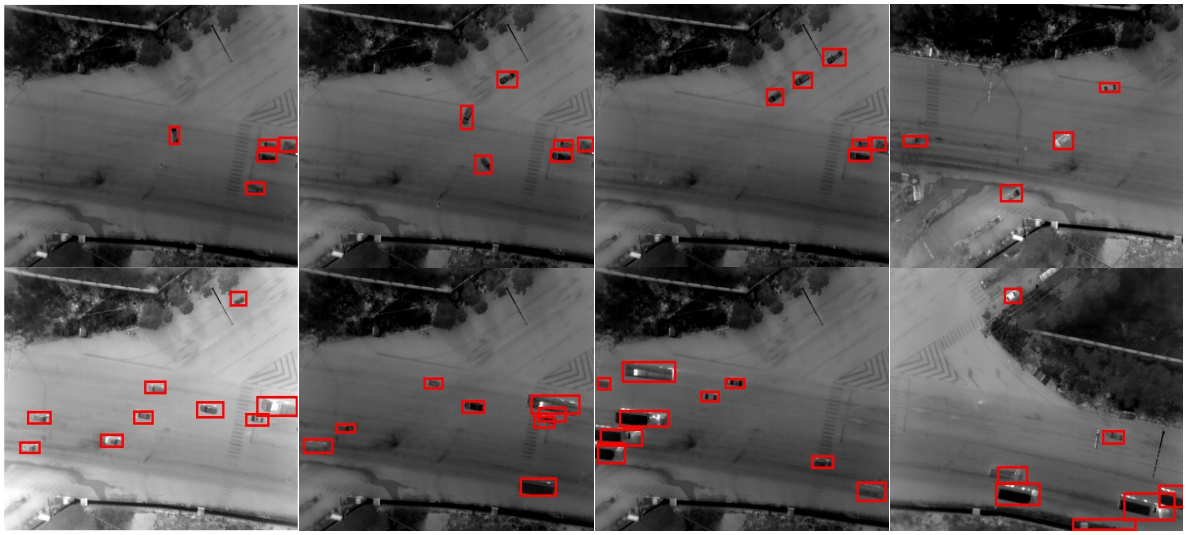

Figure 7. The performances of the method on part images of **NPU_DJM100_1** [35].

According to the analysis above, the method based on the neural network is able to accomplish vehicle detection in aerial infrared images, even in some harsh environments. The NPU_DJM100_1 and all the training samples are captured at the same time. Although we choose the NPU_DJM100_1, which is far from the the training sample in time, someone may suspect that the method may be achieved by scenario training. To remove this suspicion, we prepare additional test images (NPU_DJM100_2) captured in a scene, which are different from the scenes of the fourth times. The partial detection results are shown in Figure 8.

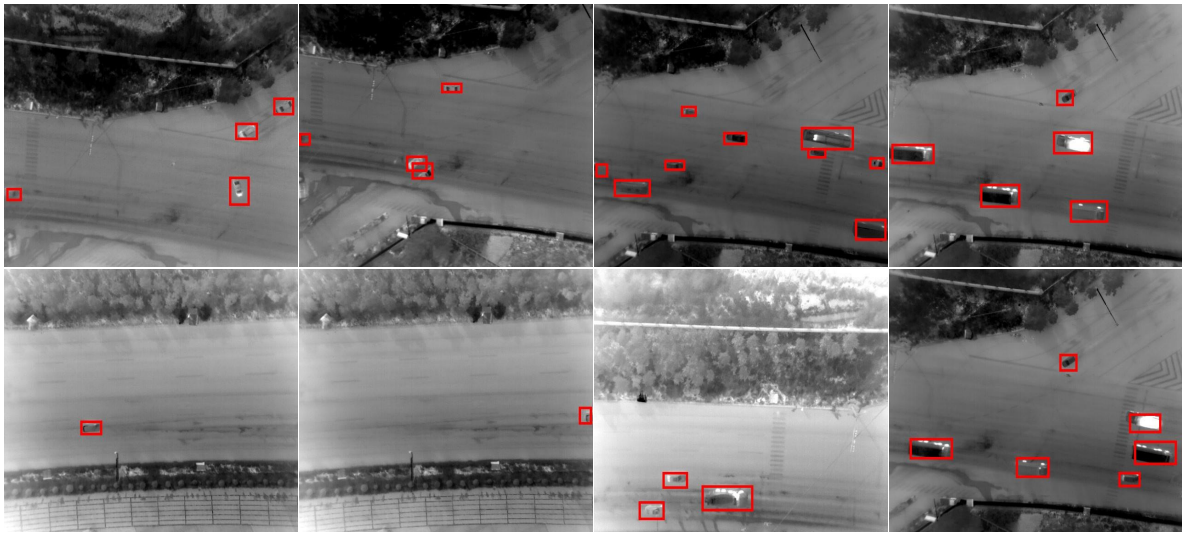

Figure 8. The performances of the method on part images of **NPU_DJM100_2** [35].

At first glance, the scenes of NPU_DJM100_2 are similar to NPU_DJM100_1 when comparing Figures 7 and 8. However, they are partly different from each other. A vehicle in the center of Figure 8 (1-4) is much brighter than all the vehicles in Figure 7 because of the blazing sunlight. The method locates the vehicle from being partly in the camera's view (1-3) to being completely in the camera's

view. Figure 7 concentrates on a crossing road, but Figure 8 focuses on the one-way traffic, especially in (2-1, 2-2, 2-3) of Figure 8. These scenarios of (2-1, 2-2, 2-3) are different from Figure 7, but the proposed method locates these vehicles appearing in those images.

Scenes Change: To further validate the expansibility of the method, we test it on some infrared images captured 80 m above the ground. The road of Scenes Change is two-way traffic. The lamp post can be clearly seen on the ground. The guard bars on the two sides of the road are exposed to high temperatures for the long term, which are similar to vehicles in brightness. This causes great disturbances for detection in the aerial infrared images.

As can be seen from Figure 9, the proposed method is capable of detecting the vehicles that run in the same or opposite directions, locating the vehicle partially when it starts to come into or escape from the view in changed scenes. This evidence above proves that the method is feasible and dependable for aerial infrared vehicle detection.

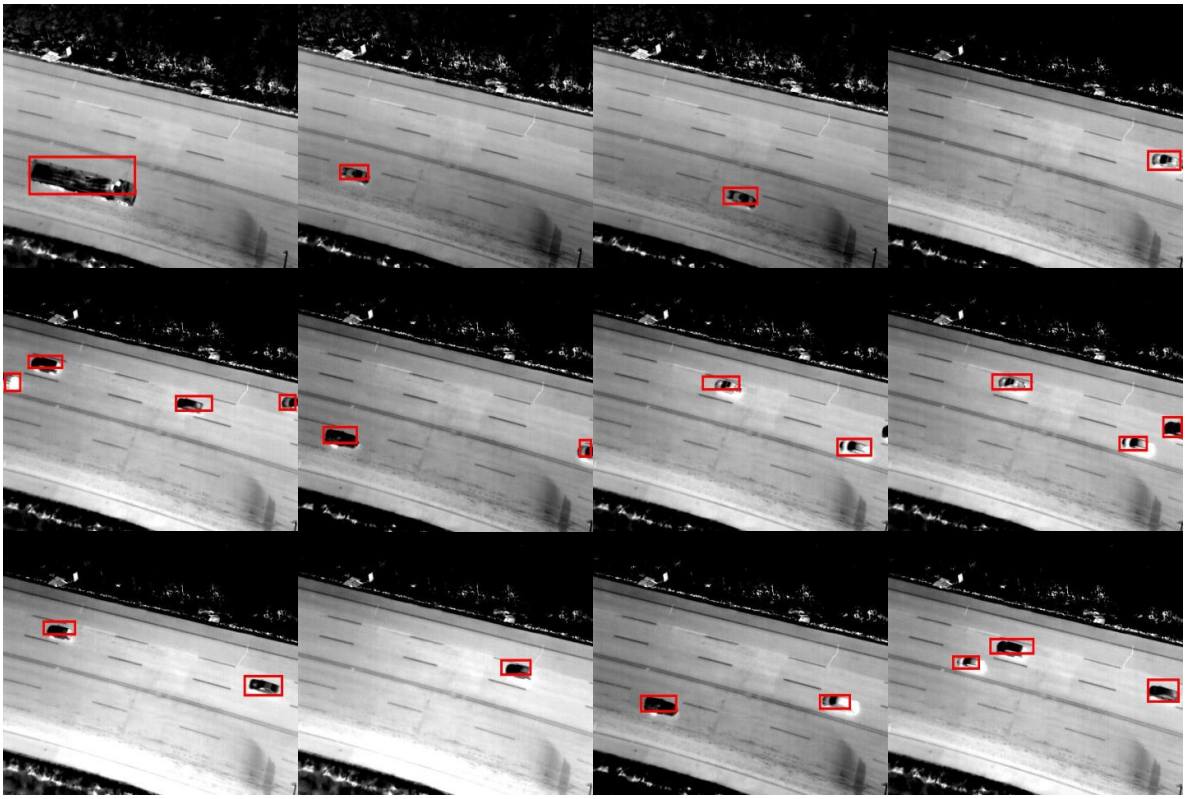

Figure 9. The performances of the method on partial images of **Scenes Change** [35].

4.2. Assessment Method

To evaluate the capability of the methodology on vehicle detection in aerial infrared images, we adopt these measurements: *Precision*, *Recall* and *F*1-*Score* defined as follows:

$$Precision = \frac{TP}{TP + FP}. \tag{15}$$

The *Precision* is the percentage of the correctly-detected vehicles' number over the total detected vehicles:

$$Recall = \frac{TP}{TP + FN}. \tag{16}$$

The *Recall* is the percentage of the correctly-detected vehicles number over the total true vehicles:

$$F1 - score = \frac{2 \times Recall \times Precision}{Recall + Precision}. \quad (17)$$

The *F*1-*Score* is a trade-off between *Recall* and *Precision*, where TP is the true positives (i.e., the number of vehicles correctly detected), FP is the false positives (i.e., the number of vehicles incorrectly detected), and FN is the false negatives (i.e., the number of other objects are wrongly regarded as vehicles).

4.3. Statistical Information

Figures 6–9 show some detection results, and then we conduct a statistical analysis about the method's performance. The details (quantitative results) are shown in Table 4.

Table 4. The performances on the test images. FP: false positive; TP: true positive; FN: false negative.

Test	Images	Vehicles	TP	FP	FN	Precision	Recall	F1-Score	Time(s)
VIVID_pktest1	100	446	388	58	7	87.00%	98.23%	92.27%	4.50
NPU_DJM100_1	189	642	612	30	22	95.33%	96.53%	95.93%	16.07
NPU_DJM100_2	190	922	903	19	30	97.94%	96.78%	97.36%	17.48
Scenes Change	100	85	79	6	0	92.94%	100%	96.34%	9.20
	Total	2095	1982	113	59	94.61%	97.11%	95.84%	

On the whole, the majority of the vehicles have been detected by the method. In total, the mean of *Precision* is 94.61%. The average of *Recall* is 97.11%. The *F*1-*Score* is basically flat . This measured information sufficiently demonstrates that the method is available for the ground vehicle detection in aerial infrared images.

4.4. Discussion

Comparison with State-of-the-Art

Figure 10 and Table 5 display a comparison about the method to a state-of-the-art method in [18]. This is a method for detecting moving objects with non-stationary cameras. It models the background through a dual-modal single Gaussian model (SGM) with age, which prevents the background model from being contaminated by the foreground pixels while still allowing the model to adapt to changes in the background, and compensates the motion of the camera by mixing neighboring models, which reduces the errors arising from motion compensation, in order to achieve rapid vehicle detection.

Table 5 illustrates that the proposed method achieves absolute advantages in *Precision*, *Recall*, and *F*1-*score* measurements. The performances of the **Scenes Change** group of [18] are equal to the proposed method, but there were very few vehicles in this group. With the vehicle's number increasing, the performance generally degrades, but the performance of the proposed is smooth and steady, maintaining a high level.

As can be seen from Figure 10, the method of [18] locates the running vehicles incorrectly, which only puts a rectangle around part of the running vehicle part even when the vehicle is completely in the image. There exist many false alarms (#77,#78) and residual errors (#97,#98,#89,#90). The proposed method is superior in the location accuracy and detection rate, which is able to detect almost all the vehicles. The red rectangles are the proposed method's detection results, the green rectangles belong to [18]. It is obvious that the detection results of [18] fluctuate drastically, especially in the # 71, #72, #75,#76 of Scenes Change.

Table 5. Comparison with the state-of-the art method, the bold value of each row is the best performance.

Test	Images Number	Vehicles	The Proposed Method			Method in [18]		
			Precision	Recall	F1-Score	Precision	Recall	F1-Score
VIVID_pktest1	1–100	446	87.00%	**98.23%**	92.27%	42.82%	77.45%	55.15%
NPU_DJM100_1	1–60	38	**100%**	**100%**	**100%**	52.63%	31.75%	39.61%
NPU_DJM100_2	1–100	501	**98.20%**	97.62%	97.91%	34.35%	40.78%	37.29%
Scenes Change	70–90	20	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
Total		1005	91.34%	92.08%	91.71%	37.98%	54.44%	44.74%

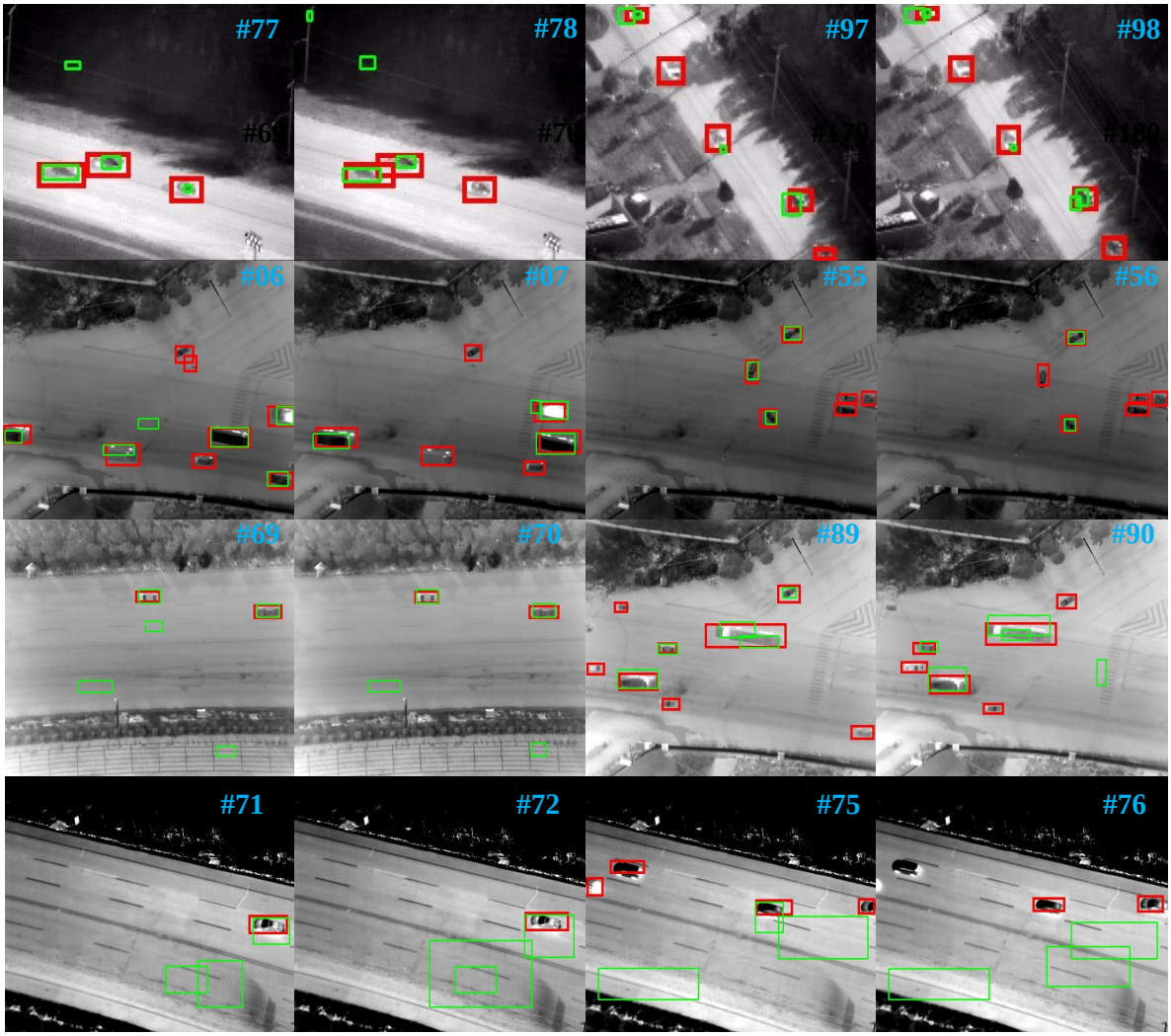

Figure 10. Comparison with a state-of-the-art method in [18]. The red rectangle is the proposed method's detection results, the green rectangles belong to [18]. From top to bottom: VIVID_pktest1, NPU_DJM100_1, NPU_DJM100_2, Scene Changes [35].

5. Conclusions

This paper proposes an efficient method for real-time ground vehicle detection in infrared imagery based on a convolutional neural network. In the proposed approach, we exploit a convolutional neural network to mine the abundant abstract features among the aerial infrared imagery. These features are more distinguished in ground vehicle detection. For ground vehicle detection, we firstly build a real-time ground vehicle detection system to capture real scene aerial images. All of the manually labeled training samples and test images are publicly posted. Then, we construct the convolutional and pooling layers and region proposal layer to achieve feature extraction. The convolutional and

pooling layers are adopted to explore the vehicle's inherent features, and the rear region proposal layer is exploited to generate candidate vehicle boxes. Finally, on the basis of a labeled sample's feature, the method iteratively learns and memorizes these features to generate a real-time ground vehicle model. It has the unique ability to detect both the stationary vehicles and moving vehicles in real urban environments. Experiments on the four different scenes demonstrate that the proposed method is effective and efficient to recognize the ground vehicles. In addition, it can accomplish the task in real time while achieving superior performances in leak and false alarm ratio. Furthermore, the current work shows great potential for ground vehicle detection in aerial imagery.

In the real world, the real-time ground vehicle detection can be applied to intelligent surveillance, traffic safety, wildlife conservation and so on. In the intelligent surveillance, the system can rapidly give the vehicle's location in imagery under day and night, which is helpful for traffic monitoring and traffic flow statistics. Traffic crashes might occur in our daily lives all the time, but it is difficult to confirm the responsibility for the accident in complex backgrounds. The system can be used to identify the principal responsible party for its real-time detection capacity. As for the wildlife conservation, most of the protected animals are caught and killed during the night. The system can locate the hunter's vehicle at night, and this helps some regulatory agencies to take countermeasures in time.

Author Contributions: X.L., T.Y. and J.L. constructed the overall system and developed the neural network. In addition, they wrote and revised the paper. X.L. and J.L. participated in the research data collection, analysis and interpretation. T.Y. guided the experiments and the statistical analysis. Additionally, they jointly designed and performed the experiments.

Acknowledgments: This research was funded by the National Natural Science Foundation of China (No. 61672429), and the ShenZhen Science and Technology Foundation (JCYJ20160229172932237).

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

Abbreviations

The following abbreviations are used in this manuscript:

SVM	Support Vector Machines
UAV	Unmanned Aerial Vehicle
SIFT	Scalar Invariant Feature Transform
HOG	Histogram of Oriented Gradient
ISM	Implicit Shape Model
NMS	Non Maximum Suppression
CNN	Convolutional Neural Network
MHI	Motion History Image
HMM	Hidden Markov Model

References

1. Diamantopoulos, G.; Spann, M. Event detection for intelligent car park video surveillance. *Real-Time Imaging* **2005**, *11*, 233–243. [CrossRef]
2. Cheng, H.; Weng, C.; Chen, Y. Vehicle detection in aerial surveillance using dynamic bayesian networks. *IEEE Trans. Image Process. A Publ. IEEE Signal Process. Soc.* **2012**, *21*, 2152–2159. [CrossRef] [PubMed]
3. Chen, H.; Zhou, Y.; Deng, C. Study and implementation of car video surveillance system. *Commun. Technol.* **2012**, *45*, 55–56.
4. Zhao, G.; Hong-Bing, M.A.; Chen, C. Design of a wireless in-car video surveillance dedicated file system. *Electron. Des. Eng.* **2016**, *24*, 10–13.
5. Bohn, B.; Garcke, J.; Iza-Teran, R.; Paprotny, A.; Peherstorfer, B.; Schepsmeier, U.; Thole, C.A. Analysis of car crash simulation data with nonlinear machine learning methods. *Procedia Comput. Sci.* **2013**, *18*, 621–630. [CrossRef]
6. Saust, F.; Wille, J.M.; Maurer, M. Energy-optimized driving with an autonomous vehicle in urban environments. In Proceedings of the Vehicular Technology Conference, Yokohama, Japan, 6–9 May 2012; pp. 1–5.

7. Sun, Z.; Bebis, G.; Miller, R. On-road vehicle detection using gabor filters and support vector machines. *Int. Conf. Digit. Signal Process.* **2002**, *2*, 1019–1022.
8. Tsai, L.; Hsieh, J.; Fan, K. Vehicle detection using normalized color and edge map. *IEEE Int. Conf. Image Process.* **2007**, *16*, 850–864. [CrossRef]
9. Zhou, J.; Gao, D.; Zhang, D. Moving vehicle detection for automatic traffic monitoring. *IEEE Trans. Veh. Technol.* **2007**, *56*, 51–59. [CrossRef]
10. Elmikaty, M.; Stathaki, T. Car detection in high-resolution urban scenes using multiple image descriptors. In Proceedings of the International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; pp. 4299–4304.
11. Yang, T.; Wang, X.; Yao, B.and Li, J.; Zhang, Y.; He, Z.; Duan, W. Small moving vehicle detection in a satellite video of an urban area. *Sensors* **2016**, *16*, 1528. [CrossRef] [PubMed]
12. Luo, P.; Liu, F.; Liu, X.; Yang, Y. Stationary Vehicle Detection in Aerial Surveillance With a UAV. In Proceedings of the 8th International Conference on Information Science and Digital Content Technology (ICIDT), Jeju, Korea, 26–28 June 2012; pp. 567–570.
13. Chen, X.; Meng, Q. Vehicle detection from UAVs by using SIFT with implicit shape model. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK, 13–16 October 2013; pp. 3139–3144.
14. Xu, Y.; Yu, G.; Wang, Y.; Wu, X.; Ma, Y. A hybrid vehicle detection method based on Viola-Jones and HOG + SVM from UAV images. *Sensors* **2016**, *16*, 1325. [CrossRef] [PubMed]
15. Kamal, S.; Jalal, A. A hybrid feature extraction approach for human detection, tracking and activity recognition using depth sensors. *Arab. J. Sci. Eng.* **2016**, *41*, 1043–1051. [CrossRef]
16. Farooq, A.; Jalal, A.; Kamal, S. Dense RGB-D map-based human tracking and activity recognition using skin joints features and self-organizing map. *Ksii Trans. Internet Inf. Syst.* **2015**, *9*, doi:10.3837/tiis.2015.05.017. [CrossRef]
17. Kamal, S.; Jalal, A.; Kim, D. Depth images-based human detection, tracking and activity recognition using spatiotemporal features and modified HMM. *J. Electr. Eng. Technol.* **2016**, *11*, 1857–1862. [CrossRef]
18. Yi, K.M.; Yun, K.; Kim, S.W.; Chang, H.J.; Jin, Y.C. Detection of moving objects with non-stationary cameras in 5.8 ms: Bringing motion detection to your mobile device. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Portland, OR, USA, 23–28 June 2013; pp. 27–34.
19. Li, L.; Zeng, Q.; Jiang, Y.; Xia, H. Spatio-temporal motion segmentation and tracking under realistic condition. In Proceedings of the International Symposium on Systems and Control in Aerospace and Astronautics, Harbin, China, 19–21 January 2006; pp. 229–232.
20. Yin, Z.; Collins, R. Moving object localization in thermal imagery by forward-backward MHI. In Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop, New York, NY, USA, 17–22 June 2006; p. 133.
21. Shao, W.; Yang, W.; Liu, G.; Liu, J. Car detection from high-resolution aerial imagery using multiple features. In Proceedings of the Geoscience and Remote Sensing Symposium, Munich, Germany, 22–27 July 2012; pp. 4379–4382.
22. Tuermer, S.; Kurz, F.; Reinartz, P.; Stilla, U. Airborne vehicle detection in dense urban areas using HOG features and disparity maps. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2013**, *6*, 2327–2337. [CrossRef]
23. Chen, Z.; Wang, C.; Luo, H.; Wang, H.; Chen, Y.; Wen, C.; Yu, Y.; Cao, L.; Li, J. Vehicle detection in high-resolution aerial images based on fast sparse representation classification and multiorder feature. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2296–2309. [CrossRef]
24. Yu, S.L.; Westfechtel, T.; Hamada, R.; Ohno, K.; Tadokoro, S. Vehicle detection and localization on bird's eye view elevation images using convolutional neural network. In Proceedings of the IEEE International Symposium on Safety, Security and Rescue Robotics, Shanghai, China, 11–13 October 2017; pp. 102–109.
25. Sommer, L.; Schuchert, T.; Beyerer, J. Fast deep vehicle detection in aerial images. In Proceedings of the Applications of Computer Vision, Santa Rosa, CA, USA, 24–31 March 2017; pp. 311–319.
26. Deng, Z.; Sun, H.; Zhou, S.; Zhao, J.; Zou, H. Toward fast and accurate vehicle detection in aerial images using coupled region-based convolutional neural networks. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2017**, *10*, 3652–3664. [CrossRef]
27. Yang, M.; Liao, W.; Li, X.; Rosenhahn, B. Vehicle detection in aerial images. *arXiv* **2018**, arXiv:1801.07339. [CrossRef]

28. Konoplich, G.V.; Putin, E.; Filchenkov, A. Application of deep learning to the problem of vehicle detection in UAV images. In Proceedings of the IEEE International Conference on Soft Computing and Measurements, St. Petersburg, Russia, 25–27 May 2016; pp. 4–6.
29. 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. [CrossRef]
30. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 142–158. [CrossRef] [PubMed]
31. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
32. Tang, T.; Zhou, S.; Deng, Z.; Zou, H.; Lei, L. Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining. *Sensors* **2017**, *17*, 336. [CrossRef] [PubMed]
33. Li, J.; Li, T.; Yang, T.; Lu, Z. Cross-domain co-occurring feature for visible-infrared image matching. *IEEE Access* **2018**, *6*, 17681–17698. [CrossRef]
34. Li, J.; Li, C.; Yang, T.; Lu, Z. A novel visual vocabulary translator based cross-domain image matching. *IEEE Access* **2017**, *5*, 23190–23203. [CrossRef]
35. Test Images and Train Samples. Available online: https://shanxiliuxiaofei.github.io/ (accessed on 2 April 2018).
36. Pktest01. Available online: http://vision.cse.psu.edu/data/vividEval/datasets/PETS2005/PkTest01/index.html (accessed on 2 April 2018).
37. LabelImg Tool. Available online: https://github.com/tzutalin/labelImg (accessed on 2 April 2018).
38. Sande, K.E.A.V.D.; Uijlings, J.R.R.; Gevers, T.; Smeulders, A.W.M. Segmentation as selective search for object recognition. In Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2012; pp. 1879–1886.
39. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
40. Everingham, M.; Gool, L.; Williams, C.; Winn, J.; Zisserman, A. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
41. Felzenszwalb, P.F.; Girshick, R.B.; Mcallester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1627–1645. [CrossRef] [PubMed]
42. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 23–28 June 2014; pp. 580–587.
43. Redmon, J. Darknet: Open Source Neural Networks in C. 2013–2016. Available online: http://pjreddie.com/darknet/ (accessed on 2 April 2018).
44. Pktest02. Available online: http://vision.cse.psu.edu/data/vividEval/datasets/PETS2005/PkTest02/index.html (accessed on 2 April 2018).