A Comparison of R-CNN and Faster R-CNN at Vehicle Detection Using the UA-DETRAC Video Dataset

Xiaofo Geng and Lulu Zha

Dublin City University, Ireland
xiaofo.geng2@mail.dcu.ie

Abstract. With the development of the city, traffic congestion has become an inevitable problem. An intelligent transportation system can make valuable suggestions for the management of urban traffic through the analysis of traffic flow. The vehicle detection based on traffic video is an important part of the traffic system. In this paper, we study the problem of image detection of vehicles with the UA-DETRAC dataset. A subset of the dataset was obtained by average sampling, and then the video images from different weather conditions were analyzed using R-CNN and Faster R-CNN object detection algorithms in deep learning. The average precision of the R-CNN in the selected dataset was 49%, with the highest precision of 58% for cloudy and the lowest precision of 37% for rainy, while the average precision of the Faster R-CNN was 71%, with the highest precision of 78% for night and the lowest precision of 58% for rainy. The study has found that weather factors have an impact on vehicle detection, which lays the foundation for improving the precision of vehicle detection in the future.

Keywords: Object detection · R-CNN · Faster R-CNN.

1 Introduction

Since 2006, there have been a lot of papers on deep neural networks published. Especially in 2012, Hinton [9] used CNN network AlexNet to win the ImageNet [2] image recognition competition in one fell swoop. Since then, neural networks have received widespread attention. At present, multiple pattern classification problems including computer vision have used deep learning. There have been many successful cases of applied deep learning in various fields now. Computer vision analysis of object movement can be roughly divided into three levels: motion segmentation, object detection; object tracking; action recognition, behavior description [15]. Among them, object detection is one of the basic tasks to be solved in the field of computer vision and in video surveillance technology.

Since the objects in the video have different poses and are often obscured, their movements are irregular. Furthermore, the depth of field, resolution, weather, lighting and other conditions of the surveillance video as well as the diversity

of the scene will affect the results of the object detection algorithm. This result directly affects subsequent tracking, motion recognition, and behavior description. Therefore, even with the development of technology today, the basic task of object detection is still a very challenging subject, and there is a lot of potential and room for improvement.

In this paper, we study the problem of vehicle detection based on the UA-DETRAC dataset [17]. There are more than 140,000 frames in the UA-DETRAC dataset [17], and a subset of the dataset was obtained through average sampling for our research. R-CNN [6] and Faster R-CNN [11] in deep learning were used as the basic object detection algorithms and the effects of video images of different weather conditions on vehicle detection were also analysed. The detection challenge results are also published by the UA-DETRAC [17], where the overall precision of the R-CNN [6] is 48.95%, the highest for sunny day is 67.52% and the lowest for rainy day is 39.06%; the overall precision of the Faster R-CNN [11] is 58.45%, the highest for night is 69.85% and the lowest for rainy day is 45.16%.

The main contributions of this paper were as follows: (1) Applied two object detection algorithms, R-CNN [6] and Faster R-CNN [11], to the UA-DETRAC dataset [17]. (2) Analyzed and compared the performance of R-CNN [6] and Faster R-CNN [11] algorithms and their possible reasons. (3) It was found that weather conditions had an influence on the effect of vehicle detection. The paper is organized as follows. In Section 2, the related work is introduced. In Section 3, the dataset used in this research is presented. Detection approaches and experimental results are presented in Section 4 and 5. We finish with a discussion and conclusions in Section 6 and 8.

2 Related Work

In 2005, the PASCAL Visual Object Classes(VOC) Challenge [3] was organized. It is a benchmark for object recognition and detection. It provides standardized datasets for object classification, and tools for accessing datasets and annotations. The competition allows to evaluate the performance of different object class recognition methods. The detection effect continues to improve. Deep learning in the early stage was restricted by algorithm theory, data, and hardware, and it did not have any advantages in terms of effect and performance. It gradually faded out of the public's vision.

In 2006, Hinton and his student Salakhutdinov [8] solved the problem of gradient disappearance in deep network training. They used unsupervised pre-training to initialize the weights and supervised training for fine-tuning. So far, the deep learning wave in academia and industry has started. In 2009, Deng et al. [2] released the ImageNet dataset, which aimed to test whether computer vision could recognize all things in nature, return to machine learning, and overcome the problem of overfitting. The ImageNet [2] is an important promoter of the development of computer vision and a key promoter of the deep learning boom, pushing object detection algorithms to new heights.

In 2012, Hinton's research team participated in the ImageNet image recognition competition [2] for the first time, proving the potential of deep learning in image recognition. They won the championship through the constructed CNN network AlexNet [9]. Because the CNN [9] won the ImageNet competition [2] in one fell swoop, it has attracted many researchers' attention. This shows that the CNN [9] has a very good effect in image recognition. Since then, computer vision has received more and more attention. Fields such as face recognition and autonomous driving have also become the focus.

Object detection is an important part of computer vision, including object positioning and classification. Object positioning is achieved by detection algorithms, which obtain regions of interest from the image. Then, the CNN [9] is used to classify the object. It is time-consuming to select a large number of proposals, and in order to solve this problem, region selection algorithms are proposed. Selective Search [14] is a popular region selection algorithm, which is used by R-CNN [6] to extract only 2,000 regions from the image, that is, region proposals. To speed up network training, Fast R-CNN [5] is proposed to provide input images directly to the CNN [9] to generate convolution feature map.

Both the R-CNN [6] and the Fast R-CNN [5] use the Selective Search [14] to find region proposals. The Selective Search [14] is a slow process which affects network performance. Therefore, Shaoqing et al. [11] proposed a new object detection algorithm Faster R-CNN, which does not use the Selective Search algorithm [14], but uses network to select region proposals. Without the limitations of the Selective Search [14], the Faster R-CNN [11] is much faster than the Fast R-CNN [5] and the R-CNN [6]. As a result, the Faster R-CNN [11] can even achieve real-time object detection.

3 Dataset

3.1 PASCAL VOC

Many excellent computer vision models such as classification, positioning, detection, segmentation, action recognition and other models are based on the PASCAL VOC Challenge [3] and its dataset launched, especially some object detection models(R-CNN, Faster R-CNN, etc.). The two most important datasets for current researchers are PASCAL VOC 2007 and PASCAL VOC 2012. These two datasets frequently appear in some current detection or segmentation papers [5, 6, 11].

Since 2007, the annual dataset of the PASCAL VOC [3] contains four major categories(vehicle, household, animal and person), a total of 20 subcategories, and only the subcategories are output when forecasting. The dataset mainly focuses on classification and detection, that is, the dataset used for classification and detection is relatively large. For other tasks such as segmentation, action recognition, etc., the dataset is generally a subset of the classification and detection dataset. The annotation information of the dataset is organized in xml files and conforms to the unified annotation standard. The PASCAL's evaluation standard is mAP(mean average precision).

The dataset folder contains Annotations, ImageSets, JPEGImages, SegmentationClass and SegmentationObject. The Annotations folder stores label files in the form of xml files. The ImageSets folder stores the segmented files of the dataset, and contains 3 subfolders(Layout, Main and Segmentation). The Main folder stores the dataset segmentation files used for classification and detection, the Layout folder is used for the person layout task, and the Segmentation folder is used for the segmentation task.

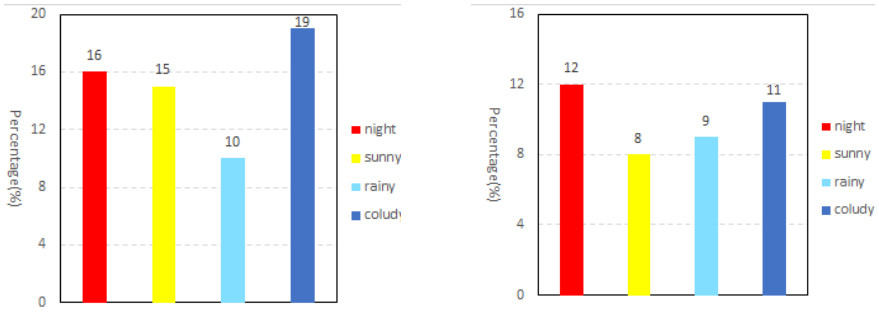
3.2 UA-DETRAC

The dataset used in our research was the UA-DETRAC dataset [17], and these sequences were selected from more than 10 hours of video captured by Cannon EOS 550D cameras in 24 different locations, representing a variety of common traffic types and conditions, including urban highways and traffic intersections [16]. Videos were recorded at 25 frames per second(fps) with a JPEG image resolution of 960×540 pixels. The dataset was divided into training and testing sets, with 83,791 and 56,340 images respectively.

Fig. 1: Some of the traffic images from the dataset

In our research, we extracted 3,600 pictures as the training and validation sets from the original training set based on the average sampling. In the original testing set, it was divided into three levels of pictures: easy, medium ,and hard based on the detection rate of the Edge-Box method [18]. We took 800 pictures from each level based on average sampling, getting a total of 2,400 images as our testing set. The following figures illustrate weather attributes in the training set and testing set(see Fig 2).

In order to facilitate the training of the models, we need to convert the UA-DETRAC dataset [17] format into the data format we need. In the R-CNN [6], because of origin annotation includes too much information which we do not use,



(a) Training set (b) Testing set

Fig. 2: Distribution of weather in training and testing data in the UA-DETRAC dataset

we extract some information which we need such as the index of the bounding box of the detected object from the xml file then put it into a txt file as the new annotation. In the Faster R-CNN [11], we converted the UA-DETRAC dataset [17] format to the PASCAL VOC dataset [3] format. First, extract the VOC format xml file of each picture from the xml file provided. Then according to the generated xml file, migrate the corresponding pictures to the target directory. Then generate trainval.txt, test.txt, train.txt and val.txt files. Replace the files with the same name in the VOC folder with these files, and the dataset format conversion is now completed.

4 Our Detection Approach

Our object detection process is based on the convolution neural network region proposal method. In this paper, the region selection and network construction are introduced in detail. All experiments were performed in the Google Colab environment. In the R-CNN [6], we first extracted regions of interest through the Selective Search algorithms [14] from the input image. These regions were then fed into a convolution neural network named VGG16 [13]. Finally, the convolution network classified the monitored regions. In the Faster R-CNN [11], the image was provided as an input to a convolution network, which provided a convolution feature map. Instead of using the Selective Search algorithms [14] on feature map to identify region proposals, a separate network was used to predict the region proposals. Then an RoI pooling layer was used to reshape the predicted region proposal, and then it was used to classify the image in the proposed region and predict the offset value of the bounding box.

4.1 Selective Search

Unlike a traditional sliding window that traverses the entire image pixel by pixel, the Selective Search algorithms [14] separate an image to identify potential ob-

jectives. Selective Search [14] is a hierarchical grouping-based algorithm, it splits images based on four attributes: color, texture, size, and shape. It is suitable for different scale images and is characterized by diversity, fast computation, and high recall [14]. The similarity is an important basis in the algorithm for segmentation and is calculated as follows.

Color Similarity Convert the color space from RGB to HSV and calculate the histogram with $bins = 25$ in each channel, so that the color histogram for each region has $25 \times 3 = 75 - dimensional$ color descriptor. The histogram is normalized by the area dimensions and the similarity is calculated using the following equation.

$$S_{color}\left(r_i, r_j\right) = \sum_{k=1}^{n} min\left(c_i^k, c_j^k\right) \tag{1}$$

where c_i^k is the histogram value for c^{th} bin in color descriptor

Texture Similarity The Gaussian distribution with a variance of 1 was used for gradient statistics in 8 directions, and then the statistical result with the size consistent with the size of the area was calculated the histogram where $bin = 10$. The result is an $8 \times 3 \times 10 = 240 - dimensional$ feature descriptor. Texture similarity of two regions is also calculated using histogram intersections.

$$S_{texture}\left(r_i, r_j\right) = \sum_{k=1}^{n} min\left(c_i^k, c_j^k\right) \tag{2}$$

where c_i^k is the histogram value for c^{th} bin in texture descriptor

Size Similarity It ensures that all scale area suggestions are formed in all parts of the image. If this factor is not taken into account, it is easy to make the merged regions continuously merge the surrounding areas, and the result is that the multi-scale is only applied to a certain part, not to the global. Therefore, we give more weight to small regions, so as to ensure that every position in the image is multi-scale merging. Size similarity is defined as:

$$S_{size}\left(r_i, r_j\right) = 1 - \frac{size(r_i) + size(r_j)}{size(im)} \tag{3}$$

where $size(im)$ is size of image in pixels.

Shape Similarity If the area r_i is included in r_j, we should merge directly. On the other hand, if r_i is difficult to connect with r_j, they will form a cliff between them, then they should not be merged. The appropriate distance of the defined area here is mainly to measure whether the two areas are more "matching". The indicator is that the smaller the Bounding Box in the merged area, the higher

the matching degree, that is, the closer the similarity is to. Shape compatibility is defined as:

$$s_{fill}(r_i, r_j) = 1 - \frac{size(BB_{ij}) - size(r_i) - size(r_j)}{size(im)} \tag{4}$$

where BB_{ij} is a bounding box around r_i and r_j.

Final Similarity The final similarity between two regions is defined as a linear combination of the aforementioned 4 similarities.

$$s(r_i, r_j) = a_1 s_{color}(r_i, r_j) + a_2 s_{texture}(r_i, r_j) + a_3 s_{size}(r_i, r_j) + a_4 s_{fill}(r_i, r_j) \tag{5}$$

where r_i and r_j are two regions in the image and $a_i \in 0, 1$.

4.2 Convolution Neural Networks

Convolution Neural Networks(CNN) is one of the representative algorithms of deep learning [7]. Convolution Neural Networks pass through multiple convolution layers and sub-sampling layers(or pooling layers), reduce the dimension of the image and extract features from it. Finally, the classifier model(fully connected layers and softmax) is used to output the final classification of the image [1]. Compared to other image classification algorithms, Convolution Neural Networks use relatively little preprocessing [4]. However, an obvious drawback of the CNN is its high memory and computational performance requirements. For example, the deep architecture of the VGG16 [13] contains approximately 138 million parameters and requires over 552M of memory [10]. In this experiment, the Convolution Neural Networks part of both the R-CNN and the Faster R-CNN used the VGG16 Network [13]. The VGG16 [13] architecture, first introduced by Karen and Andrew, consists of 13 convolution layers and 3 fully connected layers, with the largest feature that in all hidden layers, the filter is a 3×3 structure. The number of channels in filters is also regular, doubling from 64 to 128, then to 256, then to 512, while each convolution layer is followed by a pooling layer to compress the height and width of the image. It makes the VGG16 [13] neural network structure very simple, but at the same time deep enough to the number of network layers, in turn, ensures the correct classification rate.

In the R-CNN algorithm, we used transfer learning to reduce the computational load of the convolution network. After obtaining the weights of the Imagenet [2], we chose to fix the weights of the first 15 layers and modify the final softmax layer then changed its output to 2 categories, i.e., background or vehicle. The CNN structure was also used in the Faster R-CNN for the feature extraction. The structure of the whole R-CNN(Fig 3) is as follows:

- Input Layer: [$224 \times 224 \times 3$ input image layer]
- Convolution Layer $\times$ 2: [size $224 \times 224 \times 64$]
- Max Pooling Layer: [stride = 2]

- Convolution Layer × 2: [size $112 \times 112 \times 128$]
- Max Pooling Layer: [stride = 2]
- Convolution Layer × 3: [size $56 \times 56 \times 256$]
- Max Pooling Layer: [stride = 2]
- Convolution Layer × 3: [size $28 \times 28 \times 512$]
- Max Pooling Layer: [stride = 2]
- Convolution Layer × 3: [size $14 \times 14 \times 14$]
- Max Pooling Layer: [stride = 2]
- Flatten Layer: [size $= 7 \times 7 \times 512 = 25088$]
- Fully connected Layer × 2: [size = 4096]
- Softmax Layer: [size = 2]

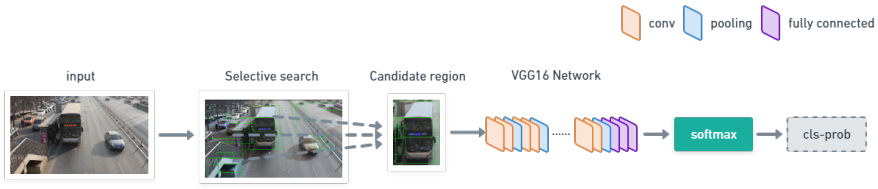


Fig. 3: R-CNN networks

4.3 Faster R-CNN Networks

The object detection networks mainly achieve two processes: the first is object positioning, i.e. to find regions of interest from the picture. The second is categorizing according to the regions of interest. The more popular way to extract proposals is Selective Search(SS) [14], which combines similar adjacent regions to get the final object region. Because the method is so time-consuming that when it is used with an efficient detection network such as Fast R-CNN [5], it is an order of magnitude slower than the Fast R-CNN [5]. If the time it takes to extract a region proposal is ignored, then the Fast R-CNN [5] can be near real-time when it is implemented using a very deep network [13]. Therefore, the region proposal algorithm is the main computing bottleneck in the object detection network [11]. Compared to the Fast R-CNN [5], Faster R-CNN [11] uses Region Proposal Networks(RPN) to generate region proposals, which can significantly speed up the extraction of region proposals.

Figure 4 shows the network structure of the Faster R-CNN. It can be seen that the Faster R-CNN consists of four main parts: Feature Extraction, RPN, Proposal Layer and RoI Pooling. These four parts will be expanded in the following sections.

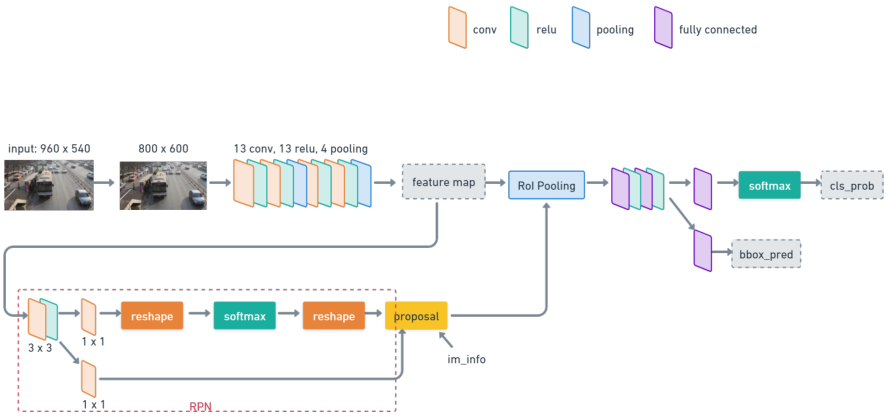


Fig. 4: Faster R-CNN networks

4.4 Feature Extraction

Feature extraction is achieved through a convolution network. In this paper, the VGG16 [13] network is used. First input a 960×540 picture, and then resize it into a fixed size 800×600 picture. Then the 800×600 picture is passed into the VGG16 network [13], which has been introduced above. As a result, the 800×600 image is fixed by the convolution network to (800/16)×(600/16). In this way, the resulting feature map can all correspond to the original picture.

4.5 RPN

The main task of RPN [11] is two: the first is to divide the feature map into multiple anchors, and then divide the anchors into positive anchors and negative anchors through the softmax classifier, i.e. to identify which anchors are foreground and which are backgrounds. The second is to get the approximate coordinates of positive anchors. At this point, the function equivalent to object positioning has been completed. The dashed red box in Figure 4 is the RPN [11], the first task RPN Classification, corresponding to the upper half of the branch, and the second task RPN Bounding Box Regression, corresponding to the lower half of the branch.

RPN Classification The RPN Classification is actually a binary classification process. For an H×W feature map, first divide the k×H×W anchors(where k = 9) evenly on the feature map, then by comparing the overlap between these anchors and ground truth to determine which anchors are foreground(IoU $>$ 0.7) and which are background(IoU $<$ 0.3), that is, to label each anchor(foreground or background). Use these labels to train the upper half of the RPN, and the RPN will have the ability to identify foregrounds and backgrounds after training.

RPN Bounding Box Regression The RPN Bounding Box Regression is used to derive a rough position of the foreground. Once anchors are labelled, we hope to adopt a way to fine-tune the positive anchors so that the positive anchors and ground truth are closer.

Suppose anchor(A_x, A_y, A_w, A_h) and ground truth(G_x, G_y, G_w, G_h), their relationship can be expressed as:

$$\begin{aligned} G_x &= A_w \cdot d_x(A) + A_x \\ G_y &= A_h \cdot d_y(A) + A_y \\ G_w &= A_w \cdot exp(d_w(A)) \\ G_h &= A_h \cdot exp(d_h(A)) \end{aligned} \tag{6}$$

$(d_x(A), d_y(A), d_w(A), d_h(A))$ is the offset between anchors and ground truth, according to formula(6), which can be expressed as:

$$\begin{aligned} d_x(A) &= (G_x - A_x)/A_w \\ d_y(A) &= (G_y - A_y)/A_h \\ d_w(A) &= log(G_w/A_w) \\ d_h(A) &= log(G_h/A_h) \end{aligned} \tag{7}$$

Use these offsets to train the lower half of the RPN, and the RPN will have the ability to identify each anchor to the corresponding optimal proposal offset after training. Note that if there are more than one ground truth in a feature map, each anchor will only select the ground truth with its highest overlap to calculate the offset.

4.6 Proposal Layer

After getting the approximate position of the proposal, it is necessary to get the accurate position of the proposal. At the end of the RPN training, the offset of anchor relative to the proposal$(d'_x(A), d'_y(A), d'_w(A), d'_h(A))$ is available, and then the approximate position of the proposal can be obtained according to formula(6). We can get many similar candidate boxes that gather around the ground truth, and then we need to select the most closest to the ground truth proposal: first select N proposals with the highest probability of the foreground, then do Non-Maximum Suppression(NMS). After that, select M proposals with the highest probability of the foreground. In order to obtain a more precise position of the proposal, it is necessary to use formula(7) to deduct how much offset there is between the proposal and the ground truth, and then return to this new offset again to complete the accurate positioning of the proposal.

4.7 RoI Pooling

There are two main tasks of the RoI Pooling Layer: the first is to extract proposals from the feature map, that is, proposal feature map. The second is to output the proposal feature map with a fixed length. Because the sizes of proposals generated by the RPN [11] are different, and the subsequent fully connected layer

used for classification must be input with a fixed length, the RoI Pooling Layer is required to fix proposals of various sizes to the same size. The previous method to solve this problem is to warp or crop the proposal, such as the R-CNN [6] and the Fast R-CNN [5]. These methods will cause image distortion and loss of information, making the classification results inaccurate. The RoI Pooling Layer circumvents this problem, and the complete proposal is more accurate for the classification results.

5 Experimental Results

For the comparison of the two object detection algorithms mentioned above, mAP(mean average precision) was used as a criterion to evaluate the algorithms. Table 1 summarizes the results. The results include the time to train and predict for one image, the validation accuracy is obtained from the training set(3,600 images) when building the model and the mAP is the average precision of prediction in the testing set(2,400 images). Given below is the analysis of the results(see Table 1).

Table 1: Comparison of performance of R-CNN and Faster R-CNN

Algorithms	**Training time**	**Prediction time**	**Validation accuracy**	**mAP**
R-CNN	7h27min	27s/Image	0.9844	0.492
Faster R-CNN	32min	0.1s/Image	0.9872	0.713

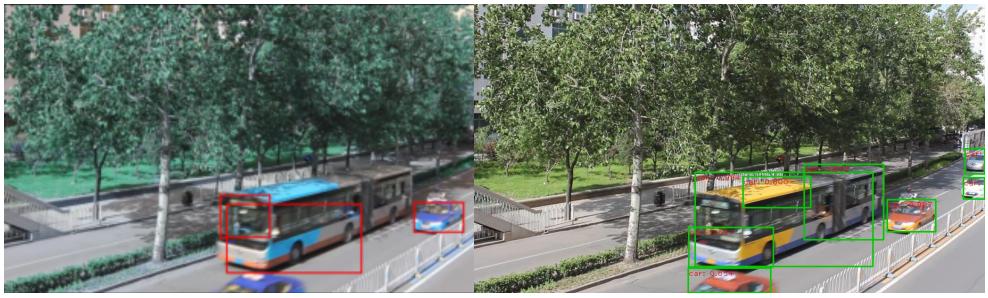
Meanwhile, we found that weather factors also had an impact on the precision of image detection. Table 2 summarizes the results. The results include the precision of the two algorithms in the four kinds of weather(sunny, cloudy, rainy and night) included in the testing set. The results are given below(see Table 2).

Table 2: Precision of vehicle detection in weather conditions

Algorithms	**Sunny**	**Cloudy**	**Rainy**	**Night**
R-CNN	0.553	0.586	0.369	0.484
Faster R-CNN	0.755	0.765	0.582	0.784

6 Discussion

As can be seen from the results, the R-CNN is less well than the Faster R-CNN both in precision and time. From the time perspective, the time cost for the Selective Search used in the R-CNN to obtain 2,000 candidate regions is about 10



(a) R-CNN (b) Faster R-CNN

Fig. 5: Results of vehicle detection in the UA-DETRAC dataset

seconds, which takes up half of the time in the image detection process. At the same time, although the R-CNN no longer uses an exhaustive strategy like the traditional sliding window approach, all the 2,000 region proposals extracted by the Selective Search require the CNN extraction features and the softmax classification. As a result, a large number of image regions are repeated calculations. The Faster R-CNN extracts features directly into the CNN from an image, and then extracts the candidate regions by the RPN after getting the feature map, which not only replaces the time-consuming Selective Search approach but also avoids the repeated calculation problem of regions, so the detection speed is greatly improved.

From the precision point of view, we mentioned that the R-CNN would put all the candidate boxes generated by the Selective Search approach, indiscriminately into the convolution network for feature extraction and classification. However, the RPN network used by the Faster R-CNN obtains positive anchors by the softmax classification and calculates the offset of bounding box regression for the anchors to obtain an accurate proposal. As well as, the NMS(Non-Maximum Suppression) algorithm is also used to reject proposals that are too small and out of bounds. Therefore, from the candidate regions that actually enter the classifier, the quality of the candidate regions generated by the RPN is higher than that of the candidate regions generated by the Selective Search.

To verify this opinion, we added a filter function after the Selective Search approach used in the R-CNN to remove candidate boxes whose proposal area is too small or whose aspect ratio is too large, and then put them into the CNN network for training. The precision of the R-CNN improved to 56% when using 400 images as the testing set, but with 2,400 images as the testing set, the precision dropped back to 48%. Thus it can be seen that the quality of the candidate regions affects the precision of the classifier in a small testing set, but this factor is not evident in front of the large testing set.

At the same time, we noticed that different weather conditions also have an impact on the precision of the object detection. In the R-CNN, the images of sunny and cloudy have far more precision than those of rainy and night. In the

Faster R-CNN, the images of rainy have far less precision than those of the other three weather conditions. We believe that this phenomenon may be caused by the light, in other words, the brightness of the image. We obtained the following results by analyzing the brightness of the testing set: sunny(131), cloudy(130), rainy(127) and night(101). From the results, it can be seen that the images of sunny, cloudy, and rainy have higher brightness than that of night, which is inconsistent with the precision of images in weather conditions. Therefore, we concluded that there is no correlation between the precision of object detection and the brightness of the image.

7 The limitations of the study

Our study is insufficient in sample size. The complete UA-DETRAC dataset contains approximately 140,000 images, but due to the RAM limitations, we cannot process all the images. Sitapa and Nattachai [12] pointed that 384GB RAM is required to run the complete dataset, which is far more than the resources we can use. So we have to choose a subset of the dataset as our sample data. This may lead to bias, so we use the average sampling method to reduce sample selection bias as far as possible.

8 Conclusion

We have compared the performance of R-CNN and Faster R-CNN for the vehicle detection problem in the UA-DETRAC dataset. Overall, the Faster R-CNN algorithm is faster and more precise than the R-CNN. A conclusion has also been come to that using the RPN instead of the Selective Search to build region proposals is the main reason for the success of the Faster R-CNN. As well as, we mentioned that weather conditions could affect the precision of the object detection algorithms. For the future work, on one hand, other different object detection algorithms can be compared(e.g. YOLO) by using the UA-DETRAC dataset. On the other hand, we intend to explore the reason why weather conditions have an effect on the precision of object detection algorithms, which will further improve the object detection performance.

References

1. Cesare Alippi, Simone Disabato, and Manuel Roveri. Moving convolutional neural networks to embedded systems: the alexnet and vgg-16 case. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 212–223. IEEE, 2018.
2. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
3. Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

4. SN Geethalakshmi. A survey on crack detection using image processing techniques and deep learning algorithms. *International Journal of Pure and Applied Mathematics*, 118(8):215–220, 2018.
5. Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
6. Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
7. Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.
8. Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.
9. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
10. Duy Thanh Nguyen, Hyun Kim, Hyuk-Jae Lee, and Ik-Joon Chang. An approximate memory architecture for a reduction of refresh power consumption in deep learning applications. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2018.
11. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
12. Sitapa Rujikietgumjorn and Nattachai Watcharapinchai. Vehicle detection with sub-class training using r-cnn for the ua-detrac benchmark. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–5. IEEE, 2017.
13. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
14. Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.
15. Sarvesh Vishwakarma and Anupam Agrawal. A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer*, 29(10):983–1009, 2013.
16. Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu. Ua-detrac: A new benchmark and protocol for multi-object detection and tracking. *arXiv preprint arXiv:1511.04136*, 2015.
17. Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *Computer Vision and Image Understanding*, 2020.
18. C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014.