applied sciences

MDPI

Article

A Deep-Learning-Based Vehicle Detection Approach for Insufficient and Nighttime Illumination Conditions

Ho Kwan Leung, Xiu-Zhi Chen, Chao-Wei Yu, Hong-Yi Liang, Jian-Yi Wu and Yen-Lin Chen *

Department of Computer Science and Information Engineering, National Taipei University of Technology, 1, Sec. 3, Chung-hsiao E. Rd., Taipei 10608, Taiwan; danielleungdaniel@gmail.com (H.K.L.); leoshiou@gmail.com (X.-Z.C.); david741002@gmail.com (C.-W.Y.); a88594562@gmail.com (H.-Y.L.); at1388072@gmail.com (J.-Y.W.)

* Correspondence: ylchen@csie.ntut.edu.tw; Tel.: +886-02-2771-2171

Received: 30 October 2019; Accepted: 7 November 2019; Published: 8 November 2019

Featured Application: This paper proposes a deep-learning-based vehicle detection technique that can achieve effective detection performance under extreme illumination conditions. The technique can be used in on-road driver assistance tools and autonomous vehicles.

Abstract: Most object detection models cannot achieve satisfactory performance under nighttime and other insufficient illumination conditions, which may be due to the collection of data sets and typical labeling conventions. Public data sets collected for object detection are usually photographed with sufficient ambient lighting. However, their labeling conventions typically focus on clear objects and ignore blurry and occluded objects. Consequently, the detection performance levels of traditional vehicle detection techniques are limited in nighttime environments without sufficient illumination. When objects occupy a small number of pixels and the existence of crucial features is infrequent, traditional convolutional neural networks (CNNs) may suffer from serious information loss due to the fixed number of convolutional operations. This study presents solutions for data collection and the labeling convention of nighttime data to handle various types of situations, including in-vehicle detection. Moreover, the study proposes a specifically optimized system based on the Faster region-based CNN model. The system has a processing speed of 16 frames per second for 500 × 375-pixel images, and it achieved a mean average precision (mAP) of 0.8497 in our validation segment involving urban nighttime and extremely inadequate lighting conditions. The experimental results demonstrated that our proposed methods can achieve high detection performance in various nighttime environments, such as urban nighttime conditions with insufficient illumination, and extremely dark conditions with nearly no lighting. The proposed system outperforms original methods that have an mAP value of approximately 0.2.

Keywords: vehicle detection; deep learning; nighttime surveillance; convolutional neural networks; insufficient lighting; ambient illumination; real-time detection; residual architecture

1. Introduction

Neural networks, convolutional neural networks (CNNs), and deep CNNs (DCNNs) have led to diverse successes in machine learning. One notable class of successes is the breakthroughs in computer vision, including image classification and object detection. Numerous CNN variants such as VGG16 [1] and ResNet101 [2] have been developed and have achieved distinctive performance in several object detection contests. Scholars have demonstrated real-time vehicle detection with object-proposal-related algorithms [3–5] based on CNNs. However, few studies have been published

regarding CNN performance levels under different levels of illumination. We tested models trained with PASCAL VOC 2007 [6], the Faster region-based CNN (R-CNN) [5] model with VGG16 [1], and ResNet101 [2] for feature extraction in low-light environments, and were unsatisfied with the performance levels.

Because the defined classes of PASCAL VOC 2007 [6] are different from our predefined classes, we could not directly compare the performance levels under the condition of insufficient lighting. We selected several images containing objects defined in PASCAL VOC 2007 [6], such as pedestrians, bicycles, and cars in urban nighttime environments and extremely dark situations. We defined the images of urban nighttime as everyday city views taken at night with rational ambient lighting. We defined extremely dark conditions as views that lacked illumination from typical sources, such as the headlights or taillights of vehicles.

Figure 1 illustrates some urban nighttime images processed by the Faster R-CNN [5] with VGG16 [1] trained using PASCAL VOC 2007 examples [6]. The bounding boxes of purple, red, and blue are for bicycles, cars, and pedestrians, respectively. This approach cannot appropriately detect occluded objects and blurry objects. The blurry outlines of objects may prevent easy identification of object features, especially at night, as mentioned in previous studies [7,8]. Figure 2 illustrates urban nighttime images processed by the Faster R-CNN [5] with ResNet101 [2] trained using the same portion of the data sets. Figure 1; Figure 2 indicate that ResNet101 [2] provides better performance than VGG16 in detecting partially shown objects. Some objects occupying a few pixels can be observed in Figure 2. However, the abilities to deal with blurry and occluded objects are still weak.

Figure 1. Urban nighttime images processed by the Faster R-CNN [5] with VGG16 [1] trained using PASCAL VOC 2007 [6].

Figure 2. Urban nighttime images processed by the Faster R-CNN [5] with ResNet101 [2] trained using PASCAL VOC 2007 [6].

Figures 3 and 4 illustrate the results of running extremely dark images through VGG16 and ResNet101 variants of the Faster R-CNN. The Faster R-CNN [5] with VGG16 [1] and ResNet101 [2] trained using PASCAL VOC 2007 [6] are unable to detect the vehicles well, whereas the optimized model trained with our nighttime images can detect them. As depicted in Figure 3, the original VGG16 [1] trained with PASCAL VOC 2007 [6] cannot recognize any objects in the row of motorbikes. As displayed in Figure 4, a system with ResNet101 [2] trained with PASCAL VOC 2007 [6] incorrectly considers an entire row of motorbikes to be a single motorbike. Both original models trained with PASCAL VOC 2007 [6] are incapable of detecting the cars located on the left-hand side under the balcony in the left image. In the present study, we aimed to address these deficiencies.

Figure 3. Extremely dark images processed by the Faster R-CNN [5] with VGG16 [1]. **Upper row**: Trained with PASCAL VOC 2007 [6]. **Lower row**: Trained with our nighttime data.

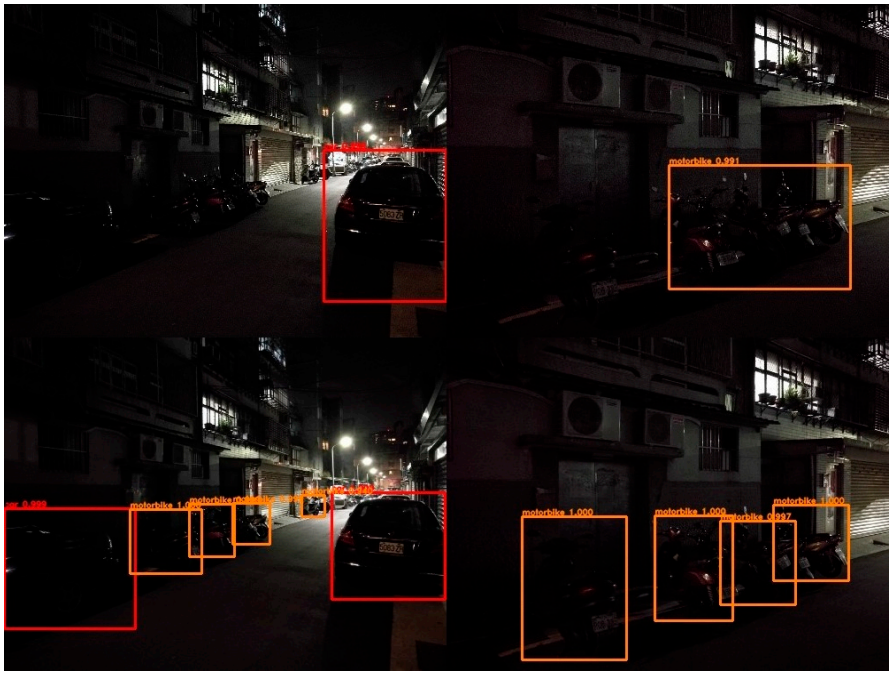

Figure 4. Extremely dark images processed by the Faster R-CNN [5] with ResNet101 [2]. **Upper row**: Trained with PASCAL VOC 2007 [6]. **Lower row**: Trained with our nighttime data.

Our predefined classes included bikes (bicycles), buses, cars, motorbikes, pedestrians, vans, and trucks. To maximize the fairness of the comparison, both our collected nighttime data classes and the PASCAL VOC 2007 classes [6] included bikes (bicycles), buses, cars, motorbikes, and pedestrians. The two selected models were the Faster R-CNN [5] with VGG16 [1] and the Faster R-CNN with ResNet101 [2]. These models were trained with PASCAL VOC 2007 [6] and tested using our collected nighttime data for the classes they had in common. Because the two data sets had different distributions and objectives, we could not directly justify the models' performance on our nighttime data. However, from Table 1, we can still obtain intuitions regarding the unsatisfactory performance levels of the models on nighttime images in the context of the effects shown in Figures 1–4. After testing the two models with two metric calculation methods, the 11-point interpolation (TRECVID) used in PASCAL VOC 2007 [6] and the method used in PASCAL VOC 2010 [9] provided a mean average precision of approximately 0.2. The model with ResNet101 [2] marginally outperformed the one with VGG16 [1]. The models are expected to perform better if the training data are collected during nighttime with enhanced labeling convention and fine adjustment aiming at feature extraction and the recognition of blurry objects.

Table 1. Performance of common classes inferred by the models trained with PASCAL VOC 2007 [6] for our nighttime data.

Classes	**Average Precision (AP)**	
	VGG16 **VOC2007 | VOC2010**	**ResNet101** **VOC2007 | VOC2010**
Bicycle (Bike)	0.1643 | 0.1330	0.1915 | 0.1570
Bus	0.1924 | 0.1642	0.2716 | 0.2442
Car	0.3787 | 0.3670	0.4251 | 0.4150
Motorbike	0.1343 | 0.1005	0.1477 | 0.1246
Pedestrian	0.1618 | 0.1131	0.1698 | 0.1244
Mean	0.2063 | 0.1756	0.2412 | 0.2130

To overcome the aforementioned difficulties in nighttime vehicle detection, this study proposes solutions for data collection, the labeling convention of nighttime image data, and a specifically optimized system based on the Faster R-CNN model [5]. As indicated by the experimental results, we obtained a mean average precision (mAP) value of approximately 0.8497. Regarding the solutions of data collection, the original models and the data sets used to train those models were incapable of achieving decent performance levels because of the characteristics of the data sets, as well as the improvements and optimizations of models, specifically for urban nighttime conditions. For extremely dark conditions, the performance levels obtained by the current models trained with the current widely used data sets were unsatisfactory. We labeled the collected data with reference to the horizontal images of size 500×375 pixels in PASCAL VOC 2007 [6]. In addition, to deal with the special exigencies of nighttime situations, several arrangements were made regarding occluded objects, blurry objects, and small objects. We conducted as much labeling as possible for occluded objects (such as rows of motorbikes). This labeling was intended to provide our model with the capability of detecting occluded objects. We labeled blurry objects as much as possible; however, we discarded a few severely unclear or invisible objects. We treated the labeling of blurry objects as a mission-critical task because our system may be deployed in vehicles, in which case, blurry or shaky images are inevitable. We provided small objects with clear definitions that related to the pixels but not the relative size. Because of the fixed behavior of convolutional operations, objects consisting of a small number of pixels may suffer from loss of features after processing in several convolutional layers. We tested and explained how feature extractors with shortcut linkages were useful for these small objects, because residual networks maintained features [2]. These data were collected from urban nighttime environments and extremely dark conditions, which allowed our system to obtain satisfactory performance in both environments, as indicated by the mAP records and subjective visual comparisons. With regard to the optimized system, we tested the ability of transfer learning for nighttime data with the Faster R-CNN [5], and selected hyperparameters by referencing the experimental results and several published studies.

In Section 2, we present several studies on object detection, traditional nighttime detection techniques, the modern method of using CNNs in nighttime detection, the treatment of the Faster R-CNN [5] for vehicle detection, data collection and labeling, and transfer learning techniques, which have relevance to our proposed methods. In Section 3, we present our proposed methods with regard to our specifically optimized system and our data collection and labeling techniques for nighttime data. In Section 4, we present our experimental results regarding how the amount of data used can affect performance, the performance with mAP records and visual comparisons in urban nighttime and extremely dark conditions, the computational efficiency, and the performance levels of different feature extractors. Finally, we present our conclusion for our entire study and contributions to nighttime vehicle detection in Section 5.

2. Related Work

Object detection is a popular research area in the field of deep learning. Scholars have presented various new object detection structures. Among the structures related to regions, the R-CNN family is popular [3–5]. R-CNN [3] uses selective search, which is a clustering algorithm for region proposal, to obtain candidates that may be predefined objects. After using selective search, R-CNN [3] uses CNNs to perform feature extraction on the selected object candidates. After the feature maps have been extracted from the previous CNNs, they are sent to the support vector machine (SVM) for classification. Although the initial bounding boxes are inaccurate, R-CNN performs regression for refining the bounding boxes so that the final bounding boxes are relatively accurate. The Fast R-CNN [4] was proposed to improve the computational efficiency of R-CNN [3]. The Fast R-CNN introduced region of interest (ROI) pooling, which provides the feature maps and the matrices representing all ROIs. ROI pooling involves the projection of ROIs to feature maps according to the imputed image. Then, the projected area is divided into sections of equal size. After ROI pooling has been conducted, the resultant maps are sent to fully connected layers for classification and bounding box regression.

The Faster R-CNN [5] is an end-to-end architecture. It dramatically improves the computational efficiency by completing the entire computational process with graphics processing unit (GPU) resources. The input images are sent to CNNs for feature extraction. Then, region proposal networks (RPNs), which are convolutional sliding windows with anchor boxes, are used to select the object candidates. The resultant maps selected from the RPNs are subjected to ROI pooling with the corresponding feature maps. After ROI pooling, the results are sent to fully connected layers for classification and bounding box regression. The CNNs used for feature selection are shared with RPNs to reduce the computational time.

CNNs are state-of-the-art structures for image recognition. VGG16 [1] is an excellent baseline CNN that performs well in several image recognition contests. Many object detection systems use VGG16 [1] for feature extraction. VGG16 can use regular 3×3 kernels to achieve satisfactory performance levels in image recognition. The receptive field of several 3×3 kernels can replace the receptive field of larger kernels. VGG16 [1] uses two sets of two 3×3 kernels and the following max pooling layer, as well as two sets of three 3×3 kernels and the following max pooling layer for convolutional feature extraction subsystems. In the object detection model, these convolutional subsystems are used as the feature extractor for later detection and the classification architecture.

Some systems use VGG16 [1]. Residual network families (ResNet) [2] are new baseline CNN models for image recognition. They use residual blocks, which are the shortcut linkages between layers, to reduce the effect of the degradation problem in deep neural networks. The residual blocks introduce a batch of identity projection that can allow the raw information to be processed through the deep layers. However, several 1×1 components exist that can increase the nonlinearity and perform dimensionality reduction and increment throughout the entire feature extraction process. ResNet101 [2] performed well in various image recognition contests, which proved that shortcut linkages are essential for increasing the depth in DCNNs. In our nighttime vehicle detection system, we must address some objects that occupy only a few pixels, as well as some blurry objects. Such challenges are inevitable in crucial situations, such as embedded systems used in moving cars. We believe that shortcut linkages can be used for dealing with small and gloomy objects after many convolutional operations, even though these convolutional operations may cause feature loss. The convolutional part of ResNet101 [2] was used as the feature extractor in our nighttime vehicle detection system.

A study [10] investigated the detailed behavior of the Faster R-CNN [5] for vehicle detection. The study indicated that the detection accuracy of the Faster R-CNN [5] increases with the size of the training images. This phenomenon occurs mainly because larger images display the main features and outlines of vehicles more clearly. In addition, the detection performance for testing images varies when an inconsistency exists between training and testing image sizes. This implies that the localization performance improves when the number of bounding boxes retained throughout the detection process is increased; however, increasing the number of bounding boxes may affect the precision. Consequently, reasonable settings for image input sizes, labeling methods, and training processes are crucial to achieve a suitable overall model performance.

Apart from the input size, labeling method, and process of training, the transfer learning technique is the most common method for training a model with relatively limited training data. In our study, we collected and labeled images by ourselves. This project did not require a large-scale collection and labeling campaign. Therefore, we used transfer learning, which initializes the model parameters with those trained in the large-scale image recognition contest. The most common transfer learning parameter is ILSVRC-2012-CLS [11], which contains millions of training data. When using their trained parameters for initialization, we must fix a certain number of neural network layers and free the rest so that the model can adapt to our collected and labeled nighttime data. For the selection of the input size and labeling method, PASCAL VOC Challenge [12] is one of the most popular methods for treating the image data. In the training and testing phases, most of the horizontal images had a size of 500×375 pixels. Images of this size can clearly display the crucial features and outlines of vehicles without excessively increasing the processing time. The labeling convention from this study

also indicates how we should label the data for detection purposes. We followed their approach and altered labels according to the behavior of nighttime images.

The traditional methods of nighttime vehicle detection applications use sensors [8] or algorithms that heavily rely on vehicles' headlights and taillights [7,13]. The authors of [7] stated that the methods based only on images depend on the headlights and taillights of the target vehicles. The algorithm is mostly suitable for the urban or city nighttime environments. Thus, a certain degree of ambient lighting, such as the street lights, is required. The method in [13] first applied traditional image preprocessing on the input frames and performed bright component information process. After that, they distinguished different cars in different lane and perform taillight clustering. Consequently, they compared current frames with previous frames in order to perform the tracking operation. Their proposed method is capable of dealing with urban and general nighttime vehicle detection; however, it is impracticable under extremely dark conditions. The technique proposed in [14] performed a global rule-based vehicle detection which contained three major steps: Bright spot segmentation, candidate taillight extraction, and candidate taillight pairing. They used an improved Otsu thresholding method, which can solve the problem of small portion of pixels of those taillights. After that, they performed the connected-component extraction in order to locate the bright spots with the optimal threshold. Then, the headlights could be paired to locate possible vehicles. However, headlight pairing may encounter a problem which it may be frustrated by two parallel motorbikes. The traditional method does not work well in extremely dark conditions with no headlights and taillights.

There are more approaches based on machine learning techniques. In [8], a satisfactory nighttime detection performance was achieved when integrating sensors and SVMs. However, multiple cues are required from vehicle lights and bodies. When the degree of illumination is poor, such as under rainy and dusky conditions, the performance may be affected. The method in [15] trained AdaBoost classifier for headlight detection in order to reduce the false detection raised by reflections; headlight pairing was also conducted, and then the paired headlights were tracked. This method was heavily relyiant on detection results of vehicles' headlights. Although using machine learning technique is a modern way to perform vehicle detection, especially in nighttime conditions, it may not obtain satisfactory detection performance under extremely dark conditions.

The modern methods of performing nighttime detection involve the use of CNNs. A study [16] on nighttime human detection with CNNs indicated that the images captured by visible-light cameras can cope well with CNNs because the neural network model can adapt to the marginal difference in the nighttime images to filter and capture the features of the object. The study used feature engineering, such as histogram equalization, to maximize the difference between the desired object and the background. However, frequently using histogram equalization may increase the processing time and we may be obliged to seek an end-to-end parallelizable algorithm that can be embedded inside vehicles with changing non-model-related configurations. Another study [17] used a visible-light camera and the Faster R-CNN [5] to deal with nighttime face detection. The study used histogram equalization before passing the images to the Faster R-CNN [5]. Because the human face is difficult to capture, one instance of the Faster R-CNN [5] was first used to detect the human body, followed by cropping of the upper body areas. Each cropped part was passed to another instance of the Faster R-CNN [5] to perform face detection. An acceptable processing speed could be reached because end-to-end models were used. To overcome the aforementioned limitations, the proposed system integrates several optimized models based on the Faster R-CNN [5], as well as an efficient data labeling scheme to achieve satisfactory performance in nighttime vehicle detection.

3. Proposed Method

3.1. Nighttime Vehicle Detection System

3.1.1. Object Detection and Feature Extractor

The Faster R-CNN [5] was selected as the fundamental objection detection algorithm for our nighttime vehicle detection system. Different pretrained CNN architectures were also used as feature extractors in the Faster R-CNN [5]. The hyperparameters were carefully selected according to a deep analysis [10] of the Faster R-CNN [5] and optimized training process. A typical Faster R-CNN [5] was used in our system. Feature extraction was performed with a CNN; objects were proposed by RPNs; ROI pooling was performed with images of different sizes; and classification was performed by fully connected layers.

We had approximately 9003 labeled nighttime images, which were insufficient for training the entire system. The system included numerous parameters that had to be trained with a large amount of data to handle the model's capacity. By using the transfer learning technique, which is a widely used method of image recognition and object detection, we could train the model without using an excessively large data set. Therefore, we used VGG16 [1] and ResNet101 [2] with parameters pretrained on ILSVRC-2012-CLS [11] to deal with the problem of an insufficient amount of training data. As shown in Figure 5, for VGG16 [1], we fixed the parameters of the first four convolutional layers with the following max pooling layers (represented in the brown part) and freed the other parts of the neural networks (as denoted in the blue part). For ResNet101 [2], we fixed the parameters of the first convolutional layer with its corresponding max pooling layers and the following single residual block (three sets of bottleneck structures), as displayed in the brown part of Figure 5. The other parameters were freed for training, as denoted in the blue part of Figure 5.

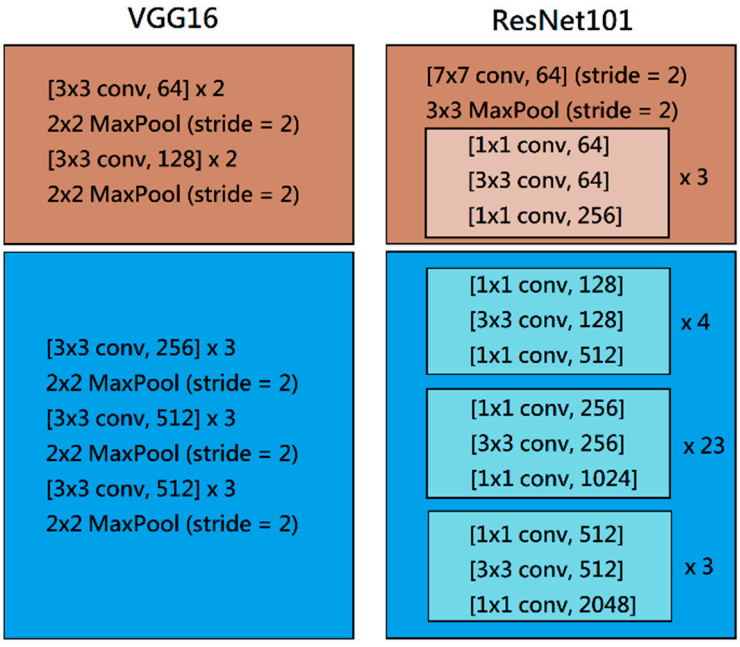

Figure 5. The usage of pre-trained convolutional neural networks (CNNs).

Although the ROI pooling in the Faster R-CNN [5] can cope with different sizes of feature maps, we fixed the input size to 500 × 375 pixels for training and inference. If the input size was larger than 500 × 375 pixels, we performed scaling. The reason for performing scaling but not allowing free size input is stated in the experimental results and discussion part. The reason is related to the challenge of objects that occupy only a few pixels. We set the number of proposed boxes before and after non-max suppression [5] during inference phase to be 2500 and 250, respectively, because a reasonably high

number of proposed boxes can help increase the recall of the system [10]. Although the accuracy is diminished when increasing the number of proposals [10], localization is vital in our nighttime vehicle detection task.

3.1.2. Lightweight Model for Embedded Systems

The aforementioned complete model requires a long time when it runs on an embedded system with low processing capability. The exact running time is stated in the experimental results and discussion. The model with ResNet101 [2] as the feature extractor required twice the running time compared with the model with VGG16 [1] for an input size of 500×375 pixels. Reducing the complexity of the model should increase the computational efficiency. However, it is recommended to suitably reduce the size of inputted images rather than the complexity of the model. Because our task did not require notably distinctive precision but recall, reducing the size of inputted images often affected the precision only.

However, certain alternative arrangements can also be made to enhance the computational efficiency of embedded systems. Considering that we do not care about the training time and only focus on the computational efficiency when generating inferences, we can reduce the number of top-scoring boxes maintained before and after the non-max suppression of RPN proposals [5]. To address concerns regarding the training time, apart from reducing the number of top-scoring boxes before and after applying non-max suppression to RPN proposals [5], the fixing of additional parameters is also a suitable strategy. However, this strategy may result in marginally lower precision and recall. For instance, we can fix one additional residual block as the feature extractor in the model when using ResNet101 [2]. When using VGG16 [1] as the feature extractor, we may fix one or two pretrained layers so that the parameters are not updated during the training phase.

3.2. Training Strategies

3.2.1. Data Set Collection Scheme and Labeling Conventions

We collected 9003 nighttime images, including images with urban nighttime and extremely dark views. These images were labeled with reference to the labeling style of the PASCAL VOC Challenge [12]. Our predefined classes included bikes, buses, cars, motorbikes, pedestrians, vans, and trucks. The real traffic data sets were collected from Taipei City, Taiwan, during nighttime. Some images were taken under slightly bright conditions or relatively dark conditions. Some images were collected with forward- and backward-driving recorders so that we could obtain both the front and back images of vehicles. For the right and left sides of vehicles, we collected data through static photo shooting. All the images were scaled from their original sizes to 500×375 pixels.

Figure 6 illustrates two labeling samples. The left image is a sample of nighttime data collected under somewhat bright conditions in an urban area, whereas the right image was collected under insufficient ambient illumination in a remote area. These two samples reveal our labeling convention in general cases, where pedestrians, motorbikes, cars, bikes, and buses are labeled using purple, red, sky-blue, deep-blue, and green boxes, respectively. Nevertheless, we still use certain special definitions for specific cases.

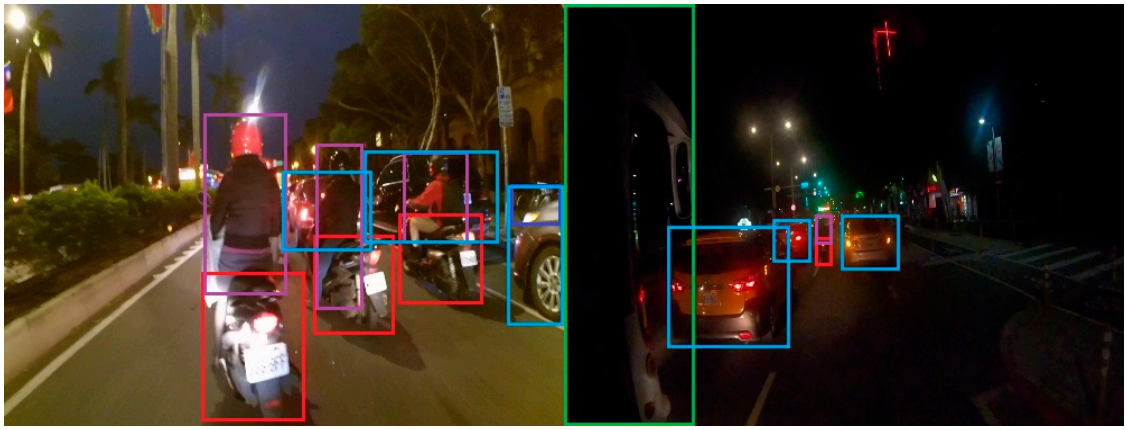

Figure 6. Two labeling samples. **Left**: Brighter nighttime image. **Right**: Darker nighttime image.

Some objects, such as a row of motorbikes parked along the roadside, are occluded. They should be labeled if their major features are distinguishable. For example, if their major parts and outlines are recognizable, then they should be labeled. We mostly did not label objects when they were blocked or covered by other objects in front of them (i.e., when the crucial features were invisible). Although many occluded objects are present in Figure 7; Figure 8, their major features and outlines are visible. They should be labeled because the instances can be distinguished. Figure 9 depicts an example of an object blocked by another object in the foreground. Although we knew that two people were riding a motorbike, we still labeled the object as one person on a motorbike because the driver's major features and outlines were blocked by the passenger. Such labeling under relatively bright images is crucial during the labeling of extremely dark images, because even the human eye finds it extremely difficult to distinguish the number of objects. Our labeling conventions focus on the visibility of major features and outlines instead of inferring whether an object is present.

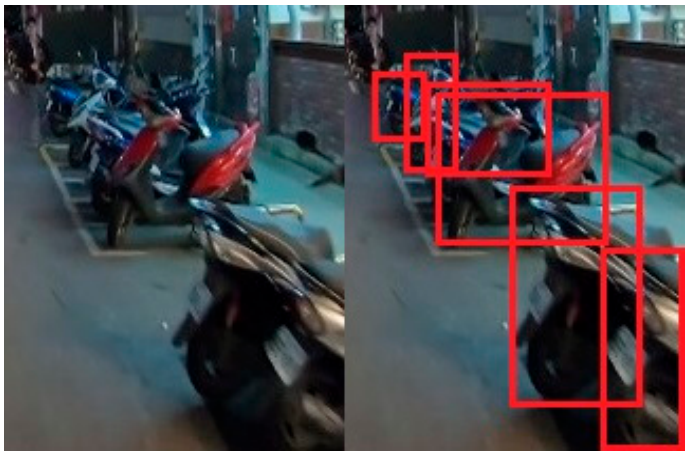

Figure 7. Occluded motorbikes with clearly visible outlines and noteworthy features. **Left**: Part cropped from unlabeled data. **Right**: Part cropped from labeled data.

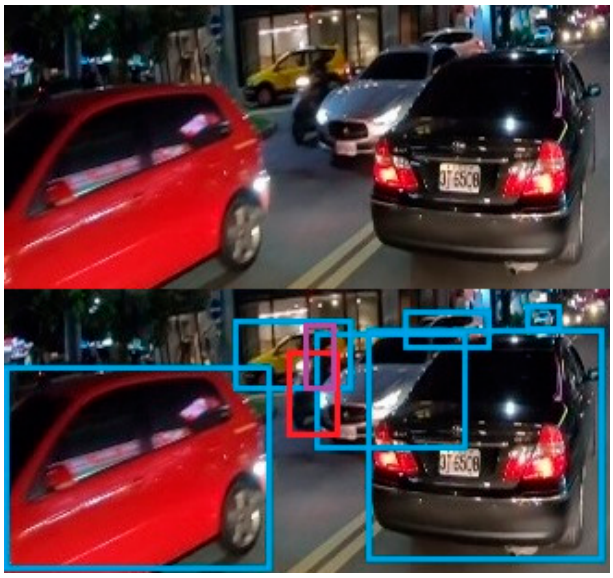

Figure 8. Occluded multiple objects with clearly visible outlines and noteworthy features. **Upper**: Part cropped from unlabeled data. **Lower**: Part cropped from labeled data.

Figure 9. Example of the major features and outlines blocked by the front objects.

Some objects occupy only a few pixels. We usually call them small objects. However, we must clearly define the term "small object" in our labeling conventions, especially in nighttime object detection. The major difficulty of nighttime detection is that the environments are lacking ambient illumination. If we include the challenges of blurry objects and occluded objects, the difficulty of nighttime detection increases. We do not call them "small objects" because, in the case of object detection, the word "small" should refer to the pixels occupied by the object. For example, an object may occupy 100×100 pixels in an image with a size of 1920×1080 pixels. We may call this object a small object. Nevertheless, it occupies a sufficient number of pixels for several convolutional operations. Therefore, we must consider the actual pixels occupied by the objects but not the sizes relative to other objects or the image sizes.

The detection performance is also related to whether the features are diminished after a certain number of traditional convolutional operations, such as VGG16 [1]. A "small" object can still be classified by a model if that object occupies a sufficient number of pixels. However, if the model involves very deep convolutional and pooling operations, a loss of crucial features and information may occur. To determine the optimal type of architecture for nighttime detection, we tested two architectures, namely VGG16 [1] and ResNet101 [2]. The related experiments are described in the sections on experimental results and discussion.

In Figure 10, the feature within the red ellipse occupies very few pixels. Thus, its noteworthy features may disappear after several downsampling operations, such as convolution and pooling. Therefore, we may not consider labeling this feature. However, the feature within the blue ellipse is more likely to be labeled.

Figure 10. Example of objects occupying only a few pixels.

After any group of images has been collected, one may find that some images are blurry. To decide whether we should discard the images or use them for the training process, we must consider the severity of blur. Directly discarding all blurry images is not a suitable method to deal with the problem of blur, because our system may be deployed inside a moving vehicle. In Figure 11, the motorbike within the orange bounding box should not be labeled because it is extremely blurry. Humans can recognize it because we know this is a photo taken on the road. However, the model only processes the area of the object (Figure 12), which is so vague that the critical features and outlines are indistinguishable.

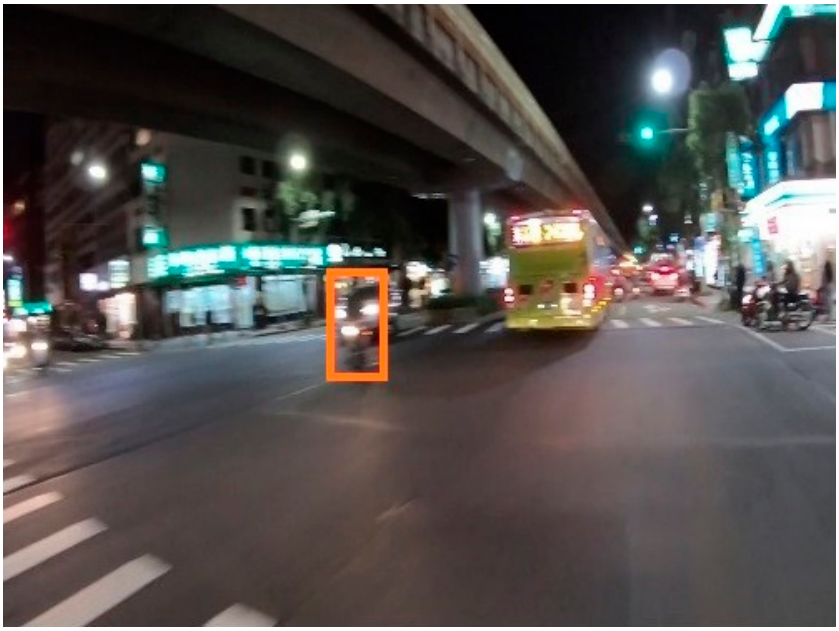

Figure 11. An example of a blurry object in a sample image.

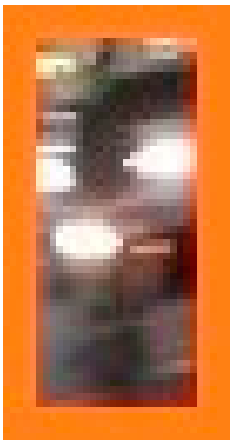

Figure 12. The blurry object from Figure 11.

3.2.2. Hyperparameter Adjustments and Optimized Training Process

Many adjustments were made and various hyperparameters were carefully selected during the development and fine-tuning of the model. A suitable training process is the key to achieving a superior nighttime vehicle detection performance.

After several experiments, we selected the initial learning rate to be 0.001, which decayed to 0.0001 after 200,000 iterations for ResNet101 [2] and 50,000 iterations for VGG16 [1]. The double learning rate was used for updating biases throughout the entire training process. This practice follows recent convention instead of the traditional standard training process. Apart from the parameters that were initialized by pretrained models, other parameters were initialized by the truncated normal distribution. The testing speed of convergence was acceptable and reasonable in our nighttime data set of real traffic conditions.

For the hyperparameters of the Faster R-CNN [5], we selected the usual setting for ROIs. The overlap threshold for an ROI was set to be 0.5 for considering the foreground. The overlap threshold for an ROI and the ground-truth box to be used as a bounding regression box was set to as 0.5. With regard to the RPN, we selected 0.7 as the intersection over union (IOU) threshold to indicate a positive example, and a threshold of 0.3 to indicate a negative example. The threshold of non-max suppression used on RPN proposals [5] was also set as 0.7. A significant number of top-scoring boxes were maintained before and after non-max suppression was applied to RPN proposals [5]. With reference to the deeper analysis [10] of the Faster R-CNN [5], we considered localization as a more crucial task than classification for nighttime vehicle detection system because finding the possible predefined objects was more important than perfectly classifying the objects. We selected 48,000 top-scoring boxes that had to be maintained before the application of non-max suppression to RPN proposals [5], and 8000 boxes after the application.

Because we disabled the RPN during testing as per the usual convention, we had to select the hyperparameters, especially the thresholds, very carefully. The overlap threshold used for non-max suppression was 0.3, and boxes with IOU values greater than or equal to the threshold were suppressed. The non-max suppression threshold used on RPN proposals was set as 0.7. The number of top-scoring boxes maintained before applying non-max suppression to RPN proposals was 2500, whereas the number maintained after applying non-max suppression was 250.

The aforementioned selection of hyperparameters was determined with reference to studies of the Faster R-CNN, especially for vehicle detection [10], and our experimental results. In our nighttime vehicle detection condition, localization takes priority, especially in the extremely dark conditions. Consequently, we focused on maintaining a higher number of bounding boxes than the thresholds before and after certain operations. Taking this as a general method of adjusting hyperparameters,

we conducted numerous experiments for searching reasonable hyperparameters to obtain satisfactory detection performance.

We evaluated different fixed parameters for transfer learning by using pretrained models of VGG16 [1] and ResNet101 [2]. We found that fixing the first residual block was a suitable strategy for our nighttime data set, because the distribution and target of data between our training nighttime data set and the data used for pretraining were quite different, which implied that the training process required more free parameters for training to achieve a superior detection performance.

4. Experimental Results and Discussion

Various experiments were conducted in this study, including the observation of the changes in the performance by models fed with different amounts of data, the searching of suitable hyperparameters, the study of the performance of the model with our collected and labeled data, some visual comparisons between the different training data sets and modes used, and the study of the model performance under extremely dark conditions.

Our real traffic nighttime data sets were collected in Taipei, Taiwan, under various road and weather conditions, including normally moving traffic, congested traffic, rainy weather, and foggy days. The illumination degree of the collected data can be classified into two categories: Urban city illumination and extremely dark conditions. For the case of urban city illumination, the data were collected at night under usual illumination, such as under street light, headlight, and taillight illumination. For the case of extremely dark conditions, the data were collected at night under extremely weak illumination. Only few lighting effects existed in these data; some of these effects could not even be distinguished by human eyes.

We collected our data sets by recording street views from dashboard cameras within the first half of the year, spanning spring and summer. After the collection of videos, we sampled the view videos with an interval of 30 seconds. Next, we viewed the sampled images and deleted images that were similar or seriously unclear before labeling them. We collected 9003 samples of data, which we labeled according to our seven predefined classes: Bikes (bicycles), buses, cars, motorbikes, pedestrians, vans, and trucks. The definition of the labels is provided in Section 3.2.1.

4.1. Different Amounts of Data Used

We recorded the changes in the mAP when models were fed with different amounts of data. In the field of deep learning, usually, the more the deep learning model is trained with training data with the right distribution for the same task, the higher is the model performance. Nevertheless, in our study related to nighttime detection, we found that this type of correlation was not explicit.

Figure 13 illustrates no explicit increase in performance despite the use of additional training data. VGG16 [1] may lack capacity of modeling the nighttime data because its performance dropped a little when additional data were used, which is an ordinary and reasonable result in this kind of convolutional neural network with the traditional architecture. The expanded data sets produced no significant increase in performance, possibly because of the high similarity of gloomy nighttime images, which is expected for the data collected under nighttime conditions. However, the capability of localizing objects can be enhanced if additional training data are used but the localizing capability cannot be reflected by the measurement of the mAP.

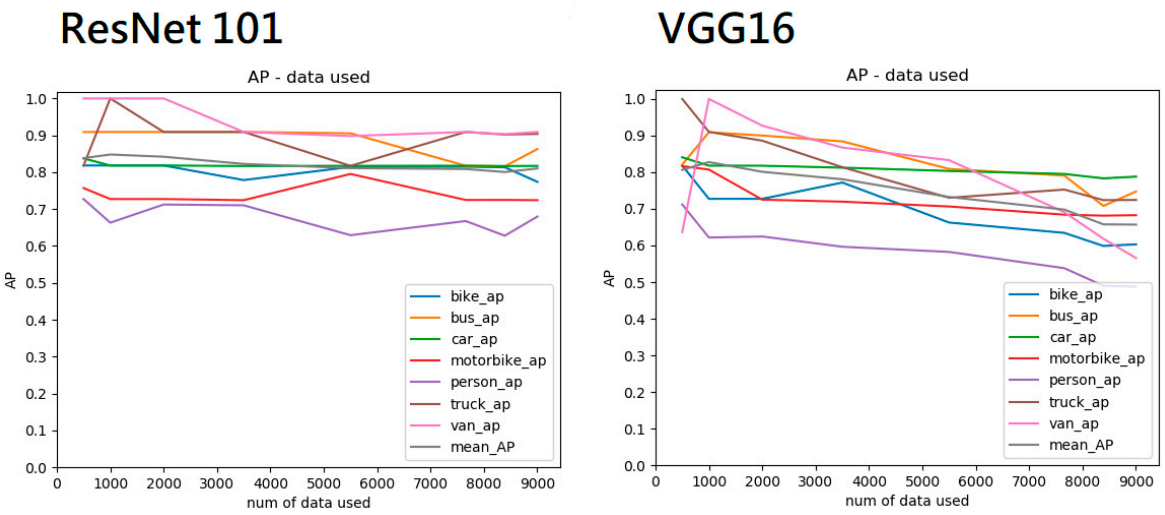

Figure 13. Changes in mean average precision (mAP) with different amounts of our nighttime data.

Figure 14; Figure 15 illustrate the performance of the Faster R-CNN [5] with VGG16 [1] and ResNet101 [2] trained using different amounts of data (500, 5500, 7659, and 9003). The model trained with 500 randomly selected nighttime images had a lower capability of localizing objects than the one trained with 9003 nighttime images. Although the performance of mAP in Figure 13 shows a stable and non-increasing trend, the visual comparisons in Figure 14; Figure 15 show the increase of the amount of our nighttime data can boost the ability of localization, which is related to the recall and cannot be measured by the mAP. Directly evaluating the model with recall as a metric is an unsuitable method because of the complex behavior required for object detection. Determining whether a bounding box is actually capturing one single predefined object is difficult. For example, determining the performance is difficult when two bounding boxes are lying on different parts of a single object, which should be bounded by one large bounding box. Knowing an object's existence and approximating the areas takes priority over inserting an exact number of bounding boxes in an image, especially for the tasks of autonomous driving and traffic surveillance.

A new metric related to recall and the IOU is required for our specific model of nighttime vehicle detection systems. This metric will be published in one of our future works. For this study, judging the ability of localization and capturing objects directly through visualization is a suitable strategy (Figure 14; Figure 15).

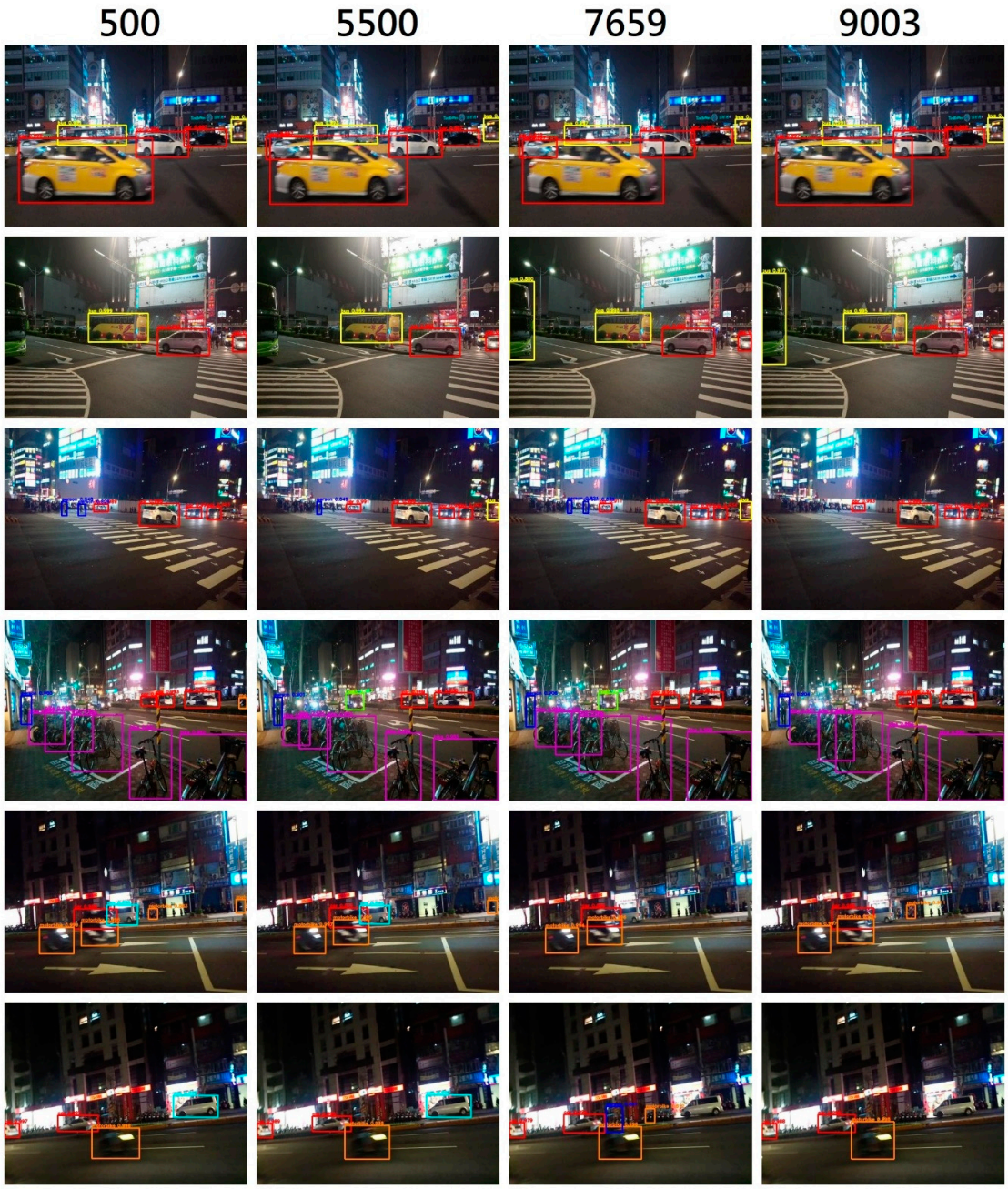

Figure 14. Faster R-CNN [5] with VGG16 [1] trained using different amounts of our nighttime data.

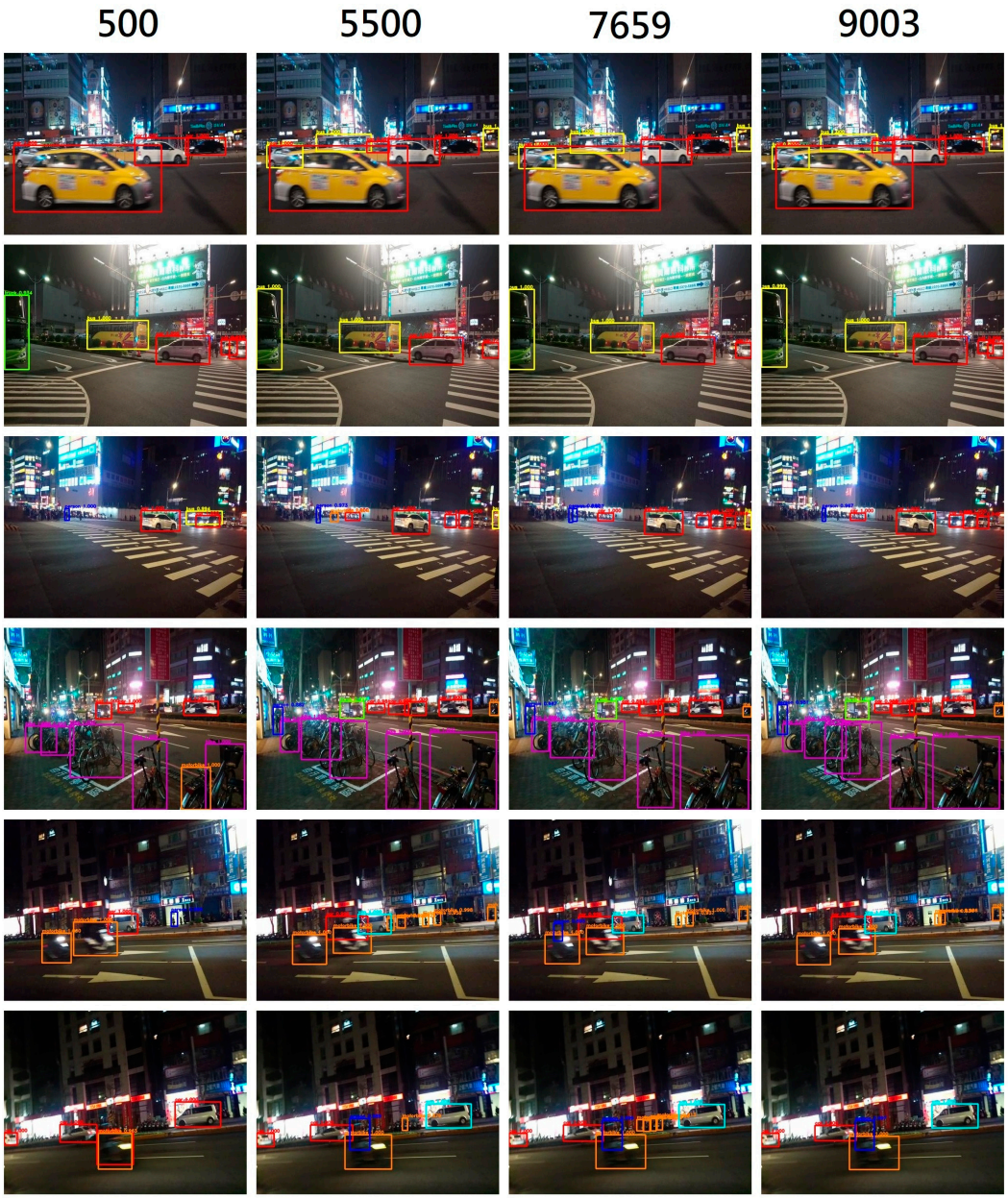

Figure 15. Faster R-CNN [5] with ResNet101 [2] trained with different amounts of our nighttime data.

4.2. Performance of Different Feature Extractors

VGG16 [1] and ResNet101 [2] were used as feature extractors for the Faster R-CNN [5]. In this section, we present the mAP measurements, training and validation loss, and visual comparisons for these two architectures.

As presented in Table 2, the mAP values of the models with VGG16 [1] and ResNet101 [2] as feature extractors were 0.7535 and 0.8497, respectively. The mAP values in the brackets of Table 2 are the performance values from Table 1, which are presented for clear and easy comparison. The performance levels were compact. No classes were predicted with extremely high or low mAP. For our nighttime case, an mAP value of approximately 0.85 was achieved. This noteworthy achievement was only possible because we labeled objects meticulously, as described in the labeling conventions (Section 3.2.1). We usually obtained a lower mAP for the class "Pedestrian" than for the other classes because of the difficulty of distinguishing a pedestrian's features with insufficient lighting background. Nevertheless, pedestrians were still labeled in our study.

Table 2. Performance of our nighttime validation data obtained using models trained with our nighttime training data.

Classes	Average Precision (AP)	
	VGG16 VOC2007 | VOC2010	**ResNet101** VOC2007 | VOC2010
Bike	0.6626 [0.1643] | 0.6815 [0.1330]	0.8099 [0.1915] | 0.8120 [0.1570]
Bus	0.8090 [0.1924] | 0.8271 [0.1642]	0.9083 [0.2716] | 0.9257 [0.2442]
Car	0.8034 [0.3787] | 0.8209 [0.3670]	0.8169 [0.4251] | 0.8625 [0.4150]
Motorbike	0.7064 [0.1343] | 0.7250 [0.1005]	0.7250 [0.1477] | 0.7857 [0.1246]
Pedestrian	0.5824 [0.1618] | 0.5799 [0.1131]	0.6714 [0.1698] | 0.6807 [0.1244]
Truck	0.7301 | 0.7726	0.9091 | 0.9356
Van	0.8329 | 0.8677	0.9091 | 0.9459
Mean	0.7324 | 0.7535	0.8214 | 0.8497

The settings of iterations for training the two aforementioned models were different because the model with ResNet101 [2] as the feature extractor required more iterations for convergence than the model with VGG16 [1] as the feature extractor. Therefore, we set 300,000 iterations for the ResNet101 model and 100,000 iterations for the VGG16 model. The models converged, as depicted in Figure 16; Figure 17.

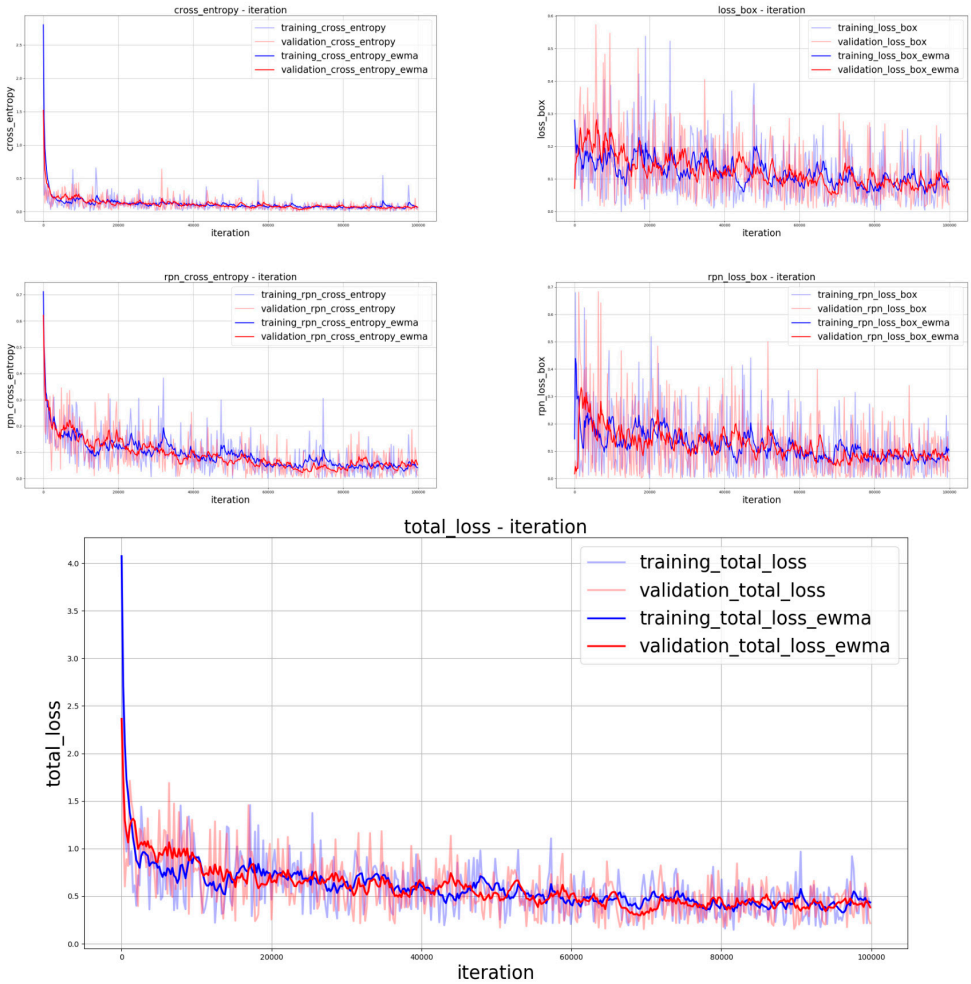

Figure 16. Training and validation loss of the Faster R-CNN [5] with VGG16 [1]. **Upper left:** Cross-entropy loss. **Middle left:** Region proposal network (RPN) cross-entropy loss. **Upper right:** Bounding box loss. **Middle right:** RPN bounding box loss. **Bottom:** Total loss.

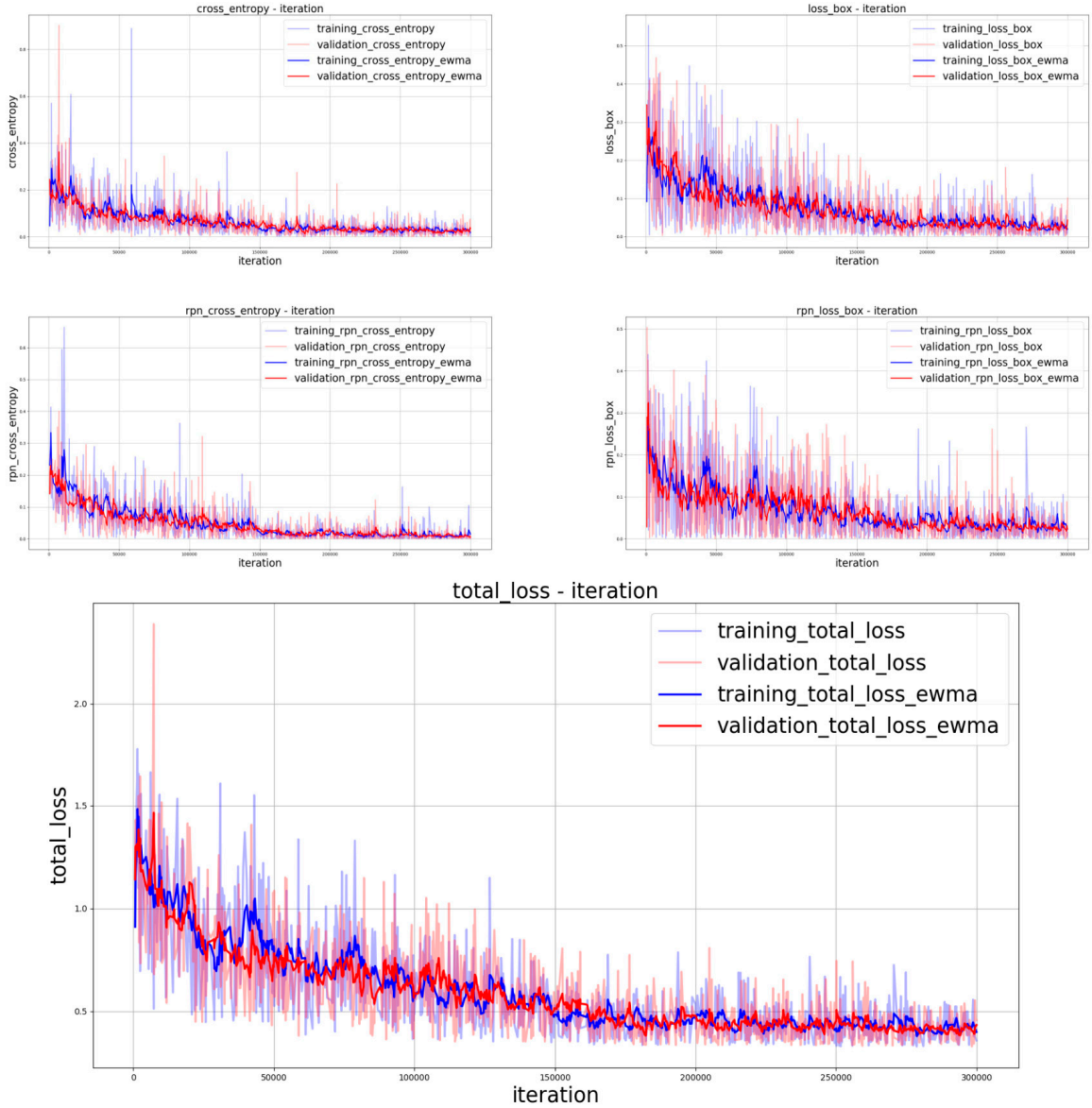

Figure 17. Training and validation loss of the Faster R-CNN [5] with ResNet101 [2]. **Upper left:** Cross-entropy loss. **Middle left:** RPN cross-entropy loss. **Upper right:** Bounding box loss. **Middle right:** RPN bounding box loss. **Bottom:** Total loss.

The reader can subjectively and visually compare outputs from ordinary models trained with PASCAL VOC 2007 [6] and outputs from our proposed systems trained with our nighttime data. The evidence indicates that our proposed system could detect objects under insufficient lighting conditions, as depicted in Figures 18–21. Models trained with our nighttime data can detect distant and blurry objects, especially the Faster R-CNN [5] with ResNet101 [2] trained using our nighttime data. This model exhibits satisfactory performance of localization and classification (lower row of Figure 20). The models trained with our nighttime data can also deal with occluded objects and objects that appear partially, such as a row of bikes or a motorbike with only half of its body in the image.

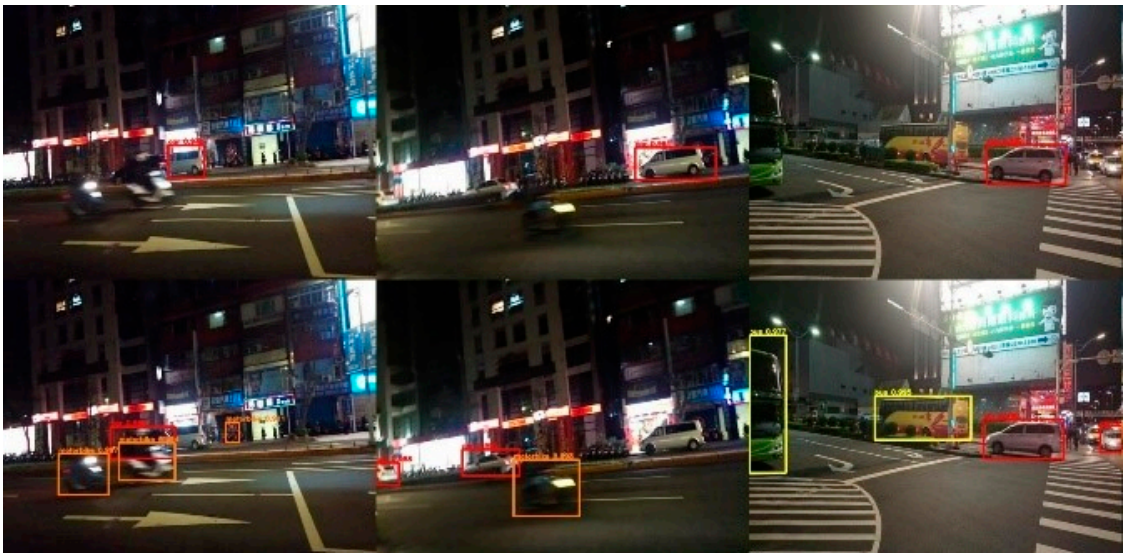

Figure 18. Output from two variants of the Faster R-CNN [5] with VGG16 [1]. **Upper row:** Ordinary model trained with PASCAL VOC 2007 [6]. **Lower row:** Proposed system trained with our nighttime data.

Figure 19. Outputs from two variants of the Faster R-CNN [5] with VGG16 [1]. **Upper row:** Ordinary model trained with PASCAL VOC 2007 [6]. **Lower row:** Proposed system trained with our nighttime data.

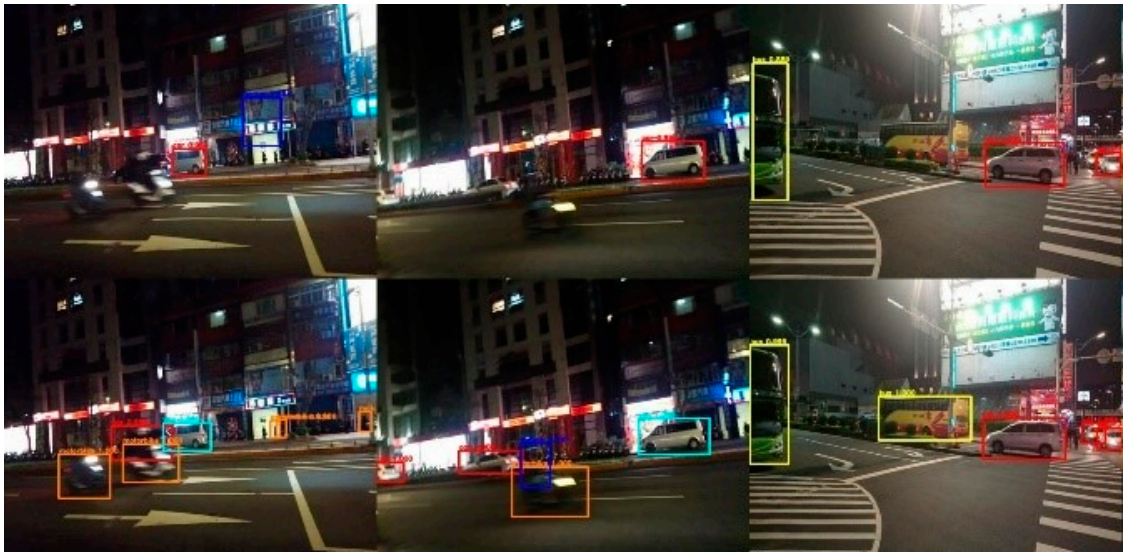

Figure 20. Outputs from two variants of the Faster R-CNN [5] with ResNet101 [2]. **Upper row:** Ordinary model trained with PASCAL VOC 2007 [6]. **Lower row:** Proposed system trained with our nighttime data.

Figure 21. Outputs from two variants of the Faster R-CNN [5] with ResNet101 [2]. **Upper row:** Ordinary model trained with PASCAL VOC 2007 [6]. **Lower row:** Proposed system trained with our nighttime data.

4.3. Computational Efficiency

The system has a different processing time when processing different sizes of images with different architectures of the feature extractor. Generally, more time is required for image processing when large images or a complex feature extractor, such as ResNet101 [2], is used. Therefore, we must scale the input size of images.

Table 3 presents the running time for different image sizes and feature extractors. The system ran on a computer with an Intel Core i7-7700K 4.20-GHz CPU, an NVIDIA GeForce GTX 1080 Ti GPU, and 24 GB of DDR4 RAM. The operating system was Ubuntu 16.04.4 LTS 64 bit. The model was built using TensorFlow 1.6 with CUDA 9.0, cuDNN 7.0, and Python 2.7. The drawing of bounding boxes and image processing parts was achieved using OpenCV 3.4.1. For an image size of 500 × 375 pixels,

which was our predefined input size, the system with VGG16 [1] as the feature extractor could run approximately 31 FPS and the system with ResNet101 could run approximately 16 FPS [2].

Table 3. Processing time of a GeForce GTX 1080 Ti graphic card for different image sizes and feature extractors.

Size of Images	Processing Time	
	VGG16	**ResNet101**
500 × 375	0.032 s	0.064 s
1000 × 750	0.079 s	0.107 s
1920 × 1080	0.151 s	0.168 s

4.4. Performance of Different Sizes of Inputted Images

The difference in the model performance for different image sizes is depicted in Figure 22; Figure 23. The depicted performance is not related to the ability to detect objects or the behavior of the models, but to the number of pixels occupied by objects. Because our models were trained with nighttime images with a size of 500 × 375 pixels, they were designed to make predictions about the usual size of objects in general 500 × 375-pixel nighttime images. Thus, the number of pixels occupied by an object in a nighttime image with a size of 500 × 375 pixels is less than that occupied by the same object in a nighttime image with a size of 1920 × 1080 pixels. Consequently, the output distribution when a large image is sent to the convolutional layers is different from that when a small image is sent to the convolutional layers, because all images are manipulated by convolutional operations with the same capacity. Therefore, the models trained with our 500 × 375-pixel nighttime images may not perform well for a large image size.

Our systems focus on the balance between the detection performance and the efficiency. If large images are used during training, the convolutional parameters are trained to deal with objects occupying a relatively large number of pixels. For example, a motorbike in a large image of 1920 × 1080 pixels may consist of 200 × 600 pixels; however, it consists of only 50 × 150 pixels in an image of 500 × 375 pixels. Because of the behavior of convolutional operations (not adapted to every size), the parameters trained with large images can only recognize an object with a pixel configuration similar to that of the training images. Any inconsistency in the training size and inference size leads to unsatisfactory performance because the convolutional layers are trained using objects with relatively large amounts of pixels. A small image depicting comparable objects may suffer from loss of features. Our method is intended for use in automobiles. Thus, our study focuses on the efficiency of in-car detection. Therefore, we referred to PASCAL VOC [6] and selected 500 × 375 pixels as the configuration for our system.

Figure 22. Inference on a 500 × 375-pixel nighttime image.

Figure 23. Inference on a 1920 × 1080-pixel nighttime image.

4.5. Performance in Extremely Dark Conditions

We measured detection performance levels for systems that processed images from extremely dark conditions under severely insufficient lighting. By using these measurements, we compared the model trained with PASCAL VOC 2007 [6] and the system trained with our nighttime data. An efficient nighttime vehicle detection system should provide satisfactory results under dark conditions. The system should detect a relevant object, even if the features of the object cannot be extracted precisely and easily. Data can be used for improving the architecture of the feature extractor and developing systems with superior performance in extremely dark conditions. The data used to train the model are an important consideration in this study.

As displayed in Figure 24; Figure 25, the proposed system trained with our nighttime data can detect relevant objects under severely insufficient lighting even if those objects (such as the row of motorbikes shown in the aforementioned figures) are typically difficult to recognize. Our system's performance levels differed from those of the models trained with the PASCAL VOC 2007 data set [6] (Figure 3; Figure 4), which may provide unfavorable localization performance. Because the photos in the PASCAL VOC 2007 data set were usually taken under sufficient lighting, the model trained with this data set could not ideally localize objects under dark conditions.

Figure 24. Outputs from the Faster R-CNN [5] with VGG16 [1] in our proposed system trained using our nighttime data.

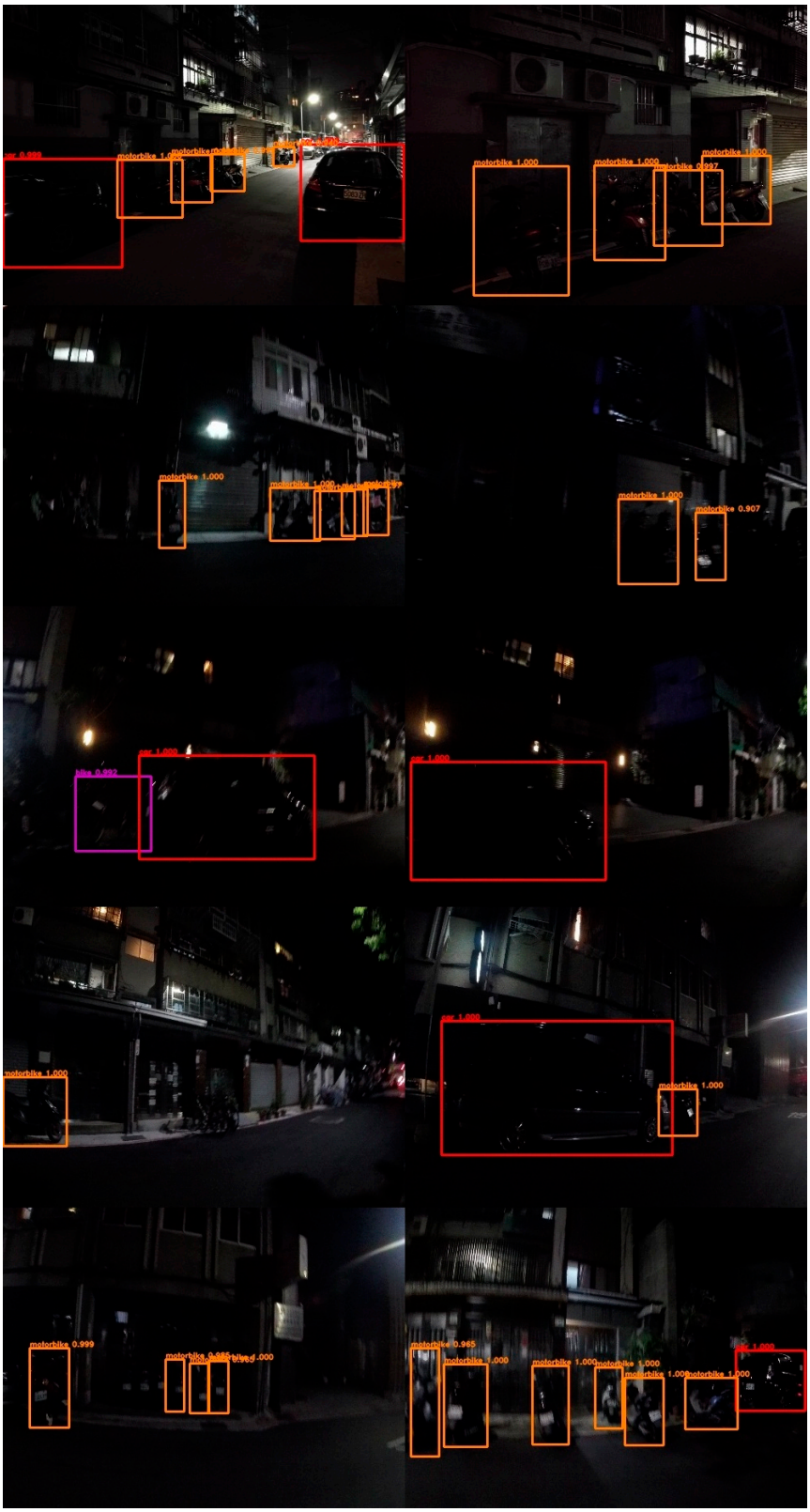

Figure 25. Outputs from the Faster R-CNN [5] with ResNet101 [2] in our proposed system trained using our nighttime data.

5. Conclusions and Future Works

In our study, we examined the nighttime detection-related behaviors of our system. Our system can tolerate severely blurred objects in images taken under urban nighttime lighting. However, for extremely dark conditions with no vehicle headlights or taillights illuminating the scene, the system can recognize the outlines of objects only if the objects are clear. Our labeling conventions pertain to occluded objects, blurry objects, and objects occupying small numbers of pixels. The results indicate that in extremely dark conditions, our innovative approach to feature extraction outperforms traditional methods [7,8], which rely heavily on the illumination from vehicles' headlights and taillights.

We also discovered that networks with shortcuts, such as ResNet101 [2], have a suitable nighttime detection performance. ResNet101 was used in our system as a feature extractor. Systems containing networks with shortcuts as feature extractors can accurately detect partially visible objects and relatively small objects during nighttime. Nevertheless, for images with a size of 500×375 pixels, the system with ResNet101 [2] required nearly twice the processing time of the system with VGG16 [1]. For embedded systems, the image size should be reduced to achieve real-time detection with a machine having relatively limited computational ability.

Our study presents labeling methods and systems optimized for occluded objects, small objects, and blurry objects, especially in extremely dark conditions. We obtained considerable improvements and optimization in terms of the performance levels. The experimental results indicate that the models trained with our nighttime data sets, which were labeled according to our conventions, could detect obscure, occluded, and small objects during nighttime or under insufficient illumination. In extremely dark conditions with nearly no illumination or extremely weak lighting, our methods provided satisfactory detection performance levels. The detection performance of the proposed methods was higher than that of the original methods. The mAP values increased from approximately 0.2 to 0.8497 with a speed of 16 FPS when processing images with a size of 500×375 pixels. This result was visually and subjectively confirmed through visual comparison of the output images. Our proposed method can effectively detect vehicles in various urban nighttime environments and under extremely dark conditions.

In our future work, we plan to improve the performance of our system by using alternative normalization methods. We plan to focus on the images taken under extreme illumination conditions. We plan to develop suitable metrics to appropriately measure the model performance for localizing objects, specifically for our nighttime data.

Author Contributions: H.K.L. contributed towards the improvement of the system, the training process, and the analysis of the system, and designed the experiments. H.K.L., X.-Z.C., and H.-Y.L. performed the experiments, conducted the experiments, analyzed the experimental data, and provided the analytical results. C.-W.Y. conducted the experiments and validations of the experimental data. H.K.L., H.-Y.L., and J.-Y.W. collected nighttime data, set up the labeling rules, and performed data labeling. H.K.L. and Y.-L.C. wrote the paper. Y.-L.C., the supervisor of H.K.L., X.-Z.C., C.-W.Y., H.-Y.L., and J.-Y.W. proofread and revised the paper, provided guidance throughout the entire preparation of the manuscript, and gave practical advice throughout the whole research process. All authors read and approved the final manuscript.

Funding: This research was funded by the Ministry of Science and Technology of Taiwan under the grant numbers MOST-107-2218-E-027 -018 and MOST-108-2221-E-027 -066.

Acknowledgments: The labeling of data for this study was partially supported by the members of the Video Coding and Transmission Lab at National Taipei University of Technology. Their contributions were greatly appreciated. This manuscript was edited by Wallace Academic Editing.

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

References

1. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.

2. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
3. 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, Columbus, OH, USA, 24–27 June 2014.
4. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015.
5. 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]
6. Everingham, M.; van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. 7 April 2007. Available online: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html (accessed on 21 June 2018).
7. Chen, Y.; Wu, B.; Huang, H.; Fan, C. A Real-Time Vision System for Nighttime Vehicle Detection and Traffic Surveillance. *IEEE Trans. Ind. Electron.* **2011**, *58*, 2030–2044. [CrossRef]
8. Chen, Y.; Chiang, H.; Chiang, C.; Liu, C.; Yuan, S.; Wang, J. A Vision-Based Driver Nighttime Assistance and Surveillance System Based on Intelligent Image Sensing Techniques and a Heterogamous Dual-Core Embedded System Architecture. *Sensors* **2012**, *12*, 2373–2399. [CrossRef] [PubMed]
9. Everingham, M.; Eslami, S.M.A.; van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. 1 March 2010. Available online: http://host.robots.ox.ac.uk/pascal/VOC/voc2010/ (accessed on 22 June 2018).
10. Fan, Q.; Brown, L.; Smith, J. A closer look at Faster R-CNN for vehicle detection. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden, 19–22 June 2016.
11. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *arXiv* **2014**, arXiv:1409.0575. [CrossRef]
12. Everingham, M.; Eslami, S.M.A.; van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
13. Kuo, Y.; Chen, H. Vision-based Vehicle Detection in the Nighttime. In Proceedings of the IEEE International Symposium on Computer, Communication, Control and Automation, Tainan, Taiwan, 5–7 May 2010.
14. Wang, J.; Sun, X.; Guo, J. A Region Tacking-Based Vehicle Detection Algorithm in Nighttime Traffic Scenes. *Sensors* **2013**, *13*, 16474–16493. [CrossRef]
15. Zou, Q.; Ling, H.; Luo, S.; Huang, Y.; Tian, M. Robusts Nighttime Vehicle Detection by Tracking and Grouping Headlights. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2838–2849. [CrossRef]
16. Kim, J.H.; Hong, H.G.; Park, K.R. Convolutional Neural Network-Based Human Detection in Nighttime Images Using Visible Light Camera Sensors. *Sensors* **2017**, *17*, 1065.
17. Cho, S.W.; Baek, N.R.; Kim, M.C.; Koo, J.H.; Kim, J.H.; Park, K.R. Face Detection in Nighttime Images Using Visible-Light Camera Sensors with Two-Step Faster Region-Based Convolutional Neural Network. *Sensors* **2018**, *18*, 2995. [CrossRef] [PubMed]