

## Article

# Detection Model of Occluded Object Based on YOLO Using Hard-Example Mining and Augmentation Policy Optimization

Seong-Eun Ryu  and Kyung-Yong Chung \* 

Division of AI Computer Science and Engineering, Kyonggi University, Suwon-Si 16227, Gyeonggi-do, Korea; fbtjddms777@kyonggi.ac.kr

\* Correspondence: dragonhci@gmail.com

**Abstract:** A study on object detection utilizing deep learning is in continuous progress to promptly and accurately determine the surrounding situation in the driving environment. Existing studies have tried to improve object detection performance considering occlusion through various processes. However, recent studies use R-CNN-based deep learning to provide high accuracy at slow speeds, so there are limitations to real-time. In addition, since such previous studies never took into consideration the data imbalance problem of the objects of interest in the model training process, it is necessary to make additional improvements. Accordingly, we proposed a detection model of occluded object based on YOLO using hard-example mining and augmentation policy optimization. The proposed procedures were as follows: diverse augmentation policies were applied to the base model in sequence and the optimized policy suitable for training data were strategically selected through the gradient-based performance improvement rate. Then, in the model learning process, the occluded objects and the objects likely to induce a false-positive detection were extracted, and fine-tuning using transfer learning was conducted. As a result of the performance evaluation, the model proposed in this study showed an mAP@0.5 value of 90.49% and an F1-score value of 90%. It showed that this model detected occluded objects more stably and significantly enhanced the self-driving object detection accuracy compared with existing model.

**Keywords:** self-driving; data mining; occlusion detection; data augmentation; transfer learning; model optimization



**Citation:** Ryu, S.-E.; Chung, K.-Y. Detection Model of Occluded Object Based on YOLO Using Hard-Example Mining and Augmentation Policy Optimization. *Appl. Sci.* **2021**, *11*, 7093. <https://doi.org/10.3390/app11157093>

Academic Editors: Sławomir Nowaczyk, Mohamed-Rafik Bouguelia and Hadi Fanaee

Received: 18 June 2021  
Accepted: 28 July 2021  
Published: 31 July 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

As the limitations of the means of transportation are significantly decreased due to the progress achieved by road traffic technology, the number of modern people who own a vehicle continues to increase, and this leads to diverse traffic problems. Of such problems, the increasing number of car accidents causes strong social risk burden to transportation users, and causes massive social and economic losses at the national level [1]. The causes of car accidents are divided into drivers, vehicles and roads. According to the statistical data analyzing car accidents, driver-related causes such as negligent driving, fatigue and poor driving account for more than 90% of the total number of car accidents [2]. Recently, as one of the solutions for reducing the number of car accidents caused by drivers' negligent driving, many studies on self-driving are being conducted. A self-driving vehicle refers to 'a vehicle that recognizes the surrounding environment and travels without an operation of a user' [3]. According to NHTSA, it is forecasted that 90% of the car accidents occurring due to driver negligence will be prevented when self-driving vehicles are commercialized [4].

As described, self-driving technology is an innovative technology that effectively reduces the number of car accidents and contributes to reducing traffic congestions and traffic rule violations. Currently, surrounding object recognition and detection are active studies, being essential technologies for commercializing self-driving. Lian, J. et al. [5] proposed a model based on the attention feature fusion method to enhance the detection

accuracy of small objects that are difficult to detect in traffic scenes. Such a model is advantageous in that it uses local and global scales to design the multi-scale channel attention block (MS-CAB) and intensively and easily recognize small objects. However, the used model was limited in that it was designed without taking into consideration occluded objects that may occur in a road environment. In addition, the model was problematic in that the data imbalance of the classes of interest to be detected was not taken into consideration in the model training process. In the real-life object detection process, the occlusion phenomenon where a part of an object is occluded by particular blocks frequently occurs. This serves as the main cause that increases the false-negative that decreases detection rate and degrades the overall detection performance [6]. In a road driving environment, a situation where a crosswalk object is undetected because it is occluded by cars or pedestrians in front is a representative example of the occlusion phenomenon that frequently occurs in the actual self-driving process. In another study on road driving detection, Huang, X. et al. [7] proposed a real-time detection and tracking model utilizing traffic image data based on the integrated two-stream convolutional neural network. This model utilizes the space stream network for object detection and the motion features for multi-object tracking to allow stable real-time object detection. However, it was still problematic in that occluded objects were not taken into consideration. In addition, since the data set used in the involved study did not take into consideration hard-negative data that induce model performance degradation, it was necessary to make additional improvements. In a road surrounding environment, diverse hard-negative examples that make it likely to incorrectly detect negative examples as positive examples exist. This increases the false-positive detection rate of a detection model and has a negative influence on the overall detection performance [8]. As described, the pre-existing studies on object detection were examined, and it was confirmed that the studies conducted by Lian, J. et al. [5], Fan, B. B. et al. [9], and Lee, S. W. et al. [10] took into consideration the hard-negative data, but were limited in that they did not take into consideration the hard-positive data and data imbalanced caused by the occlusion phenomenon. In addition, the pre-existing studies conducted by Huang, X. et al. [7], Ke, X. et al. [11], and Ha, J. et al. [12] took into consideration the data imbalance, but were limited in that they did not take into consideration the occluded objects and hard-negative objects.

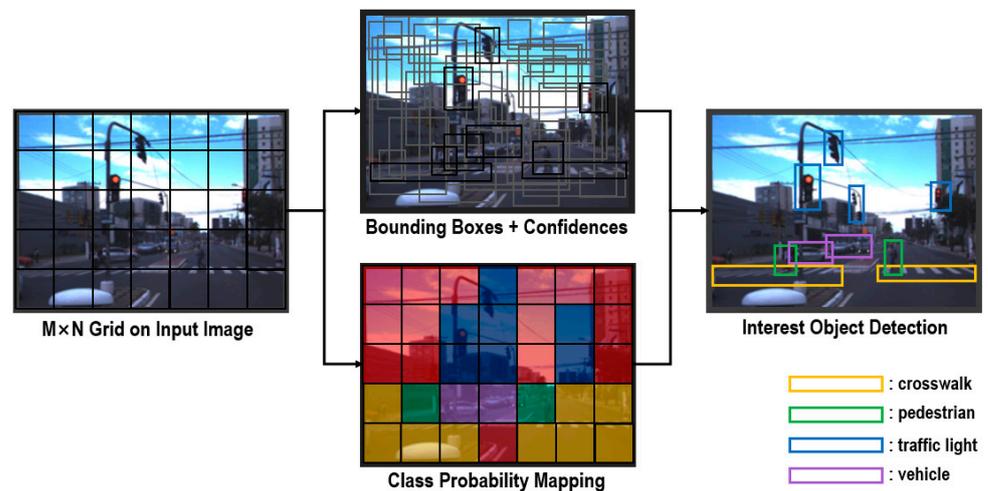
Accordingly, in this study, to bring solutions to the limitations described above, a detection model of occluded objects based on YOLO using hard-example mining and augmentation policy optimization was proposed. The proposed procedures are as follows: in the detection modeling process, the hard-positive objects occurring due to the occlusion phenomenon and the hard-negative objects that are likely to be incorrectly detected as objects of interest were extracted using the boundary feature vector according to the confidence threshold. The pre-existing learning model was additionally re-trained on the extracted feature vector through transfer learning. In addition, to overcome diverse types of data imbalance problems that may occur in the actual detection process, augmentation policies suitable for the learning model were strategically selected, and the model pre-processing optimization was conducted.

## 2. Related Work

### 2.1. Object Detection Technology Using Deep Learning

Object detection is one of the core technologies of computer vision utilized in various fields such as medicine, traffic, and health care. Object detection technology is an automated technology that separates and identifies the objects of interest from the background of an image [13]. Different from object recognition technology, which simply identifies particular objects, this technology identifies the location information of the involved objects and identifies various types of objects in one image as well as their location information. Current studies relating to object detection technology tend to focus on approaches using deep learning [13]. Deep learning algorithms such as the CNN (Convolutional Neural Network) enables a more prompt and accurate detection of the objects of interest.

Object detection algorithms using deep learning can be divided into two: one-stage detectors and two-stage detectors. One-stage detectors are detection models that promptly detect objects through the size-determined bounding box, and the size of the involved bounding box is strategically selected for application to most environments [14]. Representative one-stage detectors are the YOLO, SSD, and RetinaNet models, and these models enable prompt detection through simultaneously operating the region proposal and object distinction. Although such models are suitable for real-time detection, they are limited in that their accuracy is relatively low [14]. Two-stage detectors are models that operate the region proposal and object distinction in sequence, and such models selectively explore the regions likely to include the objects of interest through the RPN(Region Proposal Network). Representative two-stage detectors are the Faster-RCNN and R-FCN models. Although such models provide comparatively high accuracy, they are limited in that their real-time object detection processing speed is low [15]. Since the YOLO(You Only Look Once) model, the base model used in this study, has a structure of a single neural network, it is simply configured and promptly learns the surrounding information. In addition, it divides one image into grid cells and utilizes the features of the overall image from each cell. Figure 1 shows the process of the YOLO model detecting the objects of interest on the input image.



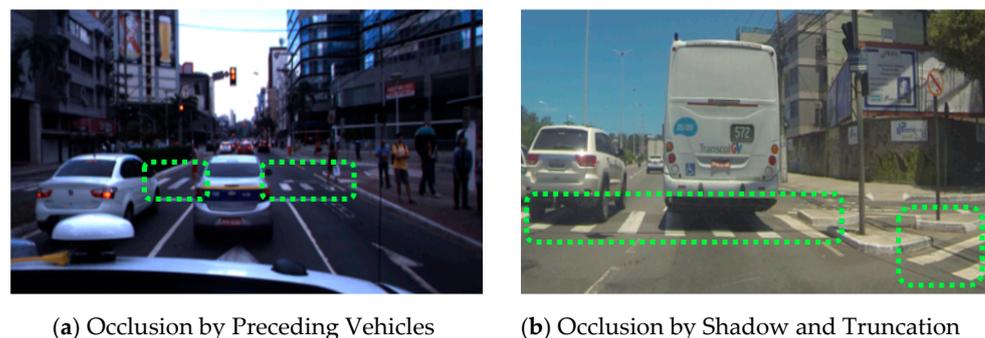
**Figure 1.** YOLO process to detect objects of interest for input images.

As shown in Figure 1, the YOLO model divides the input image into the  $M \times N$  grids and performs the end-to-end learning based on each grid cell. At this point, the bounding box and class probability within the image are deemed single regression problems [16]. This is the process of looking at the overall image once by calculating each class probability for diverse bounding boxes through the single convolutional network, and the types and location information of the objects of interest are promptly estimated through such process [17]. In the model learning process, the learned image is divided into the  $M \times N$  grid cells, and the confidence score of each cell are processed in parallel to efficiently calculate the confidence score of a particular class [18]. Through this series of prediction processes, the objects of interest to be detected by the involved model are expressed on the result image according to the class. In a study on object detection using YOLO, Jeon, S. et al. [19] proposed a real-time road driving lane detection system by extracting the features of lanes based on the YOLO model for road lane detection. This model uses the Jetson Nano Developer Kit(single board computer) and CSI-camera to collect and learn the driving data in various situations. In the involved study, the model performances were compared, and it was confirmed that the performance of the model using YOLO was relatively higher than that of the other neural network models. However, since the occlusion phenomenon caused by particular block objects was not taken into consideration in the learning process for modeling, the model was limited in that its accuracy varied

depending on the surrounding environmental factors. Accordingly, in this study, although the same YOLO model was used as the base model, an optimized model stably applicable to diverse driving environments through hard-example mining was proposed.

## 2.2. Limitations and Study Trends of Object Detection during Self-Driving

The core technologies of self-driving are divided into the recognition, determination, and control phases [20]. In the recognition phase, the multiple sensors within the self-driving vehicle are utilized to precisely identify the near and far objects. Once the objects are recognized, in the determination phase, a high-quality self-driving mechanism is implemented through the human-level surrounding environment identification and road driving learning processes. In the final control phase, the control system of the self-driving vehicle is operated based on the recognition and determination results, and actions suitable for that particular situation are taken. At this point, in the process of understanding the road traffic situation and surrounding facilities, object detection technology using deep learning is applied. The recognition process using object detection during self-driving results in different actions on the road depending on the performance differences [21]. This is a really important process, since it has a direct influence on road safety. Accordingly, a number of studies are being actively conducted to improve the object recognition and detection performances. Aslani, S. et al. [22] proposed an optic flow estimation algorithm based object detection and tracking process by taking into consideration the moving direction and size of the feature points of objects among frames in the image input phase. This model uses a median filter to effectively remove diverse types of noises in a traffic situation. However, since the involved study is a process based on the image sequence of static camera images, it is difficult to detect the prompt changes made in a continuously and dynamically changing self-driving environment. In addition, since the occluded objects were not taken into consideration in the process of detecting, filtering and dividing the objects of interest from the static image sequence, the model was limited in that the detection performance may be significantly degraded when the objects of interest are partially occluded by other objects. In the process of using the median filter, occlusion objects are likely to be judged as noise, and there is a point to be improved that occlusion objects do not properly detect them as objects of interest. During real-life road driving, the occlusion phenomenon frequently occurs and serves as the main factor that degrades the object detection performance. Figure 2a,b are road driving images included in the IARA data set and in the GOPRO data set, respectively, and shows examples of the occlusion phenomenon [23,24].



**Figure 2.** Occlusion by various road driving environments.

Figure 2a shows the occlusion phenomenon where it is difficult to detect the crosswalk, because it is occluded by the vehicles in front. Figure 2b shows the occlusion phenomenon where the crosswalk located at the center is occluded by the shadow of the vehicles in front. In addition, due to the limited size of the image input from the self-driving vehicle, the crosswalk located to the right is cut and only partially shown, making it an occluded object that is difficult to detect. As described, in a self-driving situation, there is a problem that occluded objects that are difficult to detect occur depending on diverse

road environmental conditions such as climate and traffic volume. Ning, C. et al. [25] released a pedestrian detection survey to consider occlusion. The results of the survey show that modern method-based algorithms using deep learning are more powerful than conventional method-based detection algorithms that manually select pedestrian features. Nevertheless, existing deep learning-based object detection algorithms to solve occlusion problems have several issues to improve. The first is the training data problem, which, when the amount of data is small, does not provide a good detection effect with current algorithms. This can be overcome by optimizing data augmentation to increase the amount and quality of training data. The second is the problem of accuracy and speed, and in most detection algorithms, detection accuracy and detection speed have a trade-off relationship, so it is important to design an efficient algorithm with both accuracy and detection speed depending on its application. Kulkarni, R. et al. [26] proposed a faster R-CNN-based object detection model using transfer learning. This modeling method based on pre-trained transfer learning reduces the time complexity required to develop a neural network model. However, because a faster R-CNN, a two-stage detector, is used as the base model, the method is limited in that the real-time object detection processing speed is relatively low. In addition, since the data pre-processing optimization is not taken into consideration in the model learning process for detecting the objects of interest, additional challenges relating to that problem remain. Among the most common data imbalance problems is the foreground-to-background imbalance problem, which refers to the presence of extreme imbalances between the number of positive examples and negative examples. This causes over-fitting and is a major factor in reducing detection accuracy. Therefore, various top-down and bottom-up approaches must be considered to improve this. As described, a number of self-driving related studies are being continuously conducted to detect the objects of interest under diverse environments and conditions.

### **3. Detection Model of Occluded Object Based on YOLO Using Hard-Example Mining and Augmentation Policy Optimization**

In this study, a detection model of occluded object based on YOLO using hard-example mining and augmentation policy optimization was proposed to detect objects that occlude a crosswalk on a real-time basis. The proposed model has a structure of a three-phase process. Figure 3 shows the process configuration diagram of a YOLO-based occlusion object detection model. As shown in Figure 3, the first phase is the augmentation policy optimization phase. As far as the training data for occluded object detection are concerned, the self-driving image dataset provided from the IARA(Intelligent Robotic Autonomous Automobile) and GOPRO was collected [23,24]. The above dataset includes diverse objects that appear in a road driving environment. It also contains high-quality image datasets that include various times, spaces, and climate environments. In this study, the involved data were used to create a model that detects objects that occlude a crosswalk. Nine augmentation policies were applied to the collected data to create new training data. The augmentation policies consisted of MixUp, CutOut, CutMix, Mosaic, Blurring, Brightness, Contrast, Hue, and GrayScale [27]. These policies were applied to the base YOLO model learning in sequence, and extracted the bounding box regression value of the objects of interest calculated based on the IoU(Intersection over Union) index. The actual ground truth value and predicted regression value were compared, and only the policies having a positive influence on occluded crosswalk object detection were accepted through the gradient-based residual.

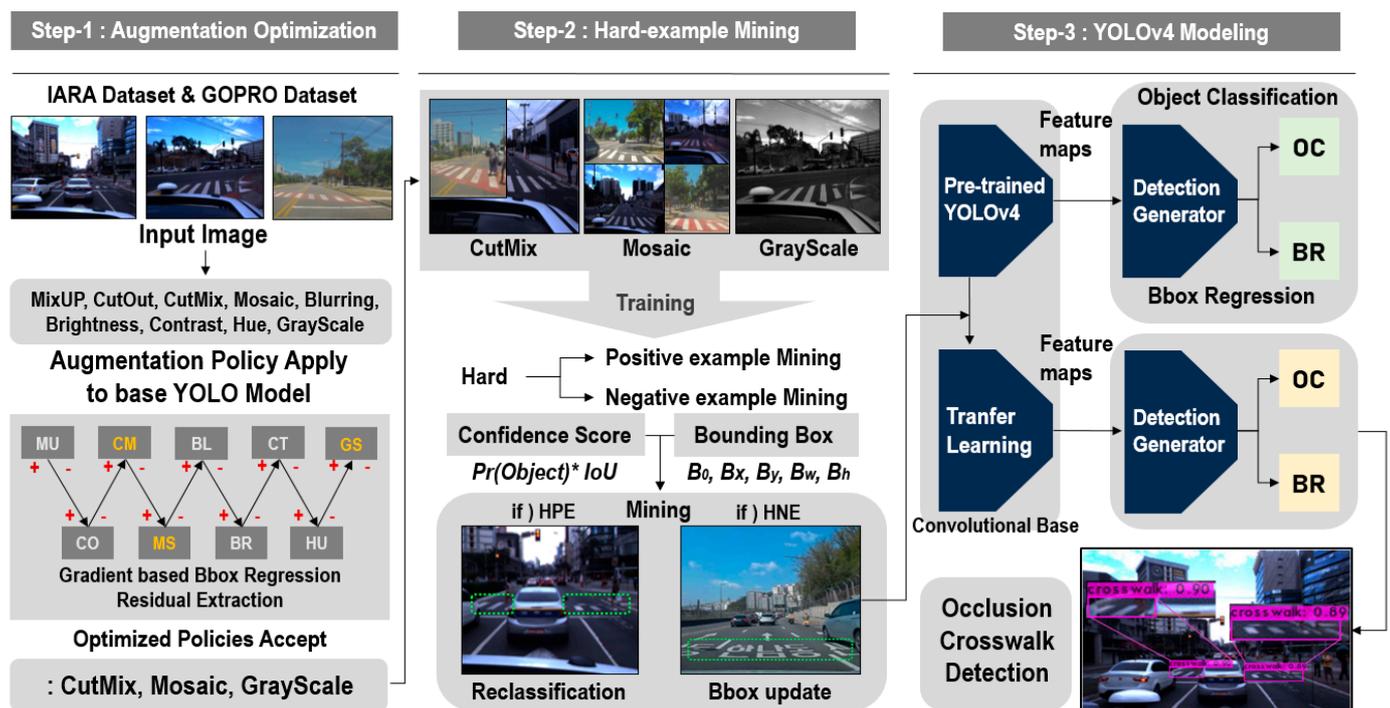


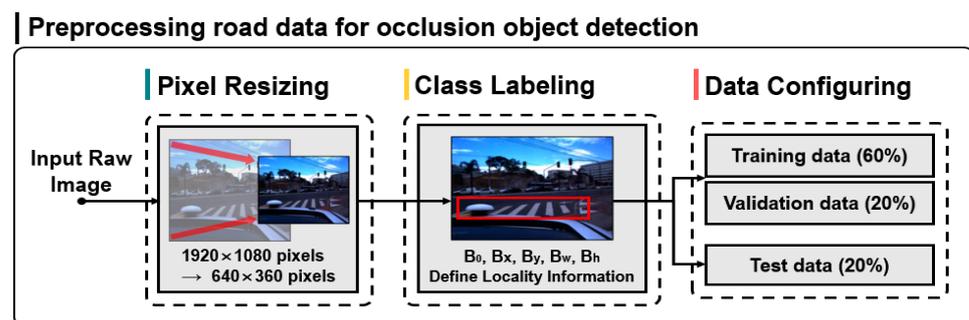
Figure 3. Process configuration diagram of YOLO based occlusion object detection model.

In the second phase, the pre-existing training data and the data to which the optimized augmentation policy is applied were used to extract the hard-positive example and hard-negative example groups through hard-example mining. Through hard-example mining, the data increasing the false-negative detection rate and false-positive detection rate based on the feature vector were extracted from the confidence score and bounding box parameters. As far as the hard-positive data increasing the false-negative detection rate are concerned, the reclassification of the objects of interest was performed. As far as the hard-negative data increasing the false-positive detection rate are concerned, the Bbox information of the false-positive objects was updated and applied to the detection model. In the final phase, a neural network is configured based on the YOLOv4 model, a one-stage detector, to detect the objects that occlude a crosswalk. The augmentation policies applied through the previous phases were optimized, and fine-tuning was conducted through hard-example re-learning. This is the process of applying transfer learning to the pre-trained detection model, and has a positive influence on occluded object detection. The occluded crosswalk was detected based on the object classification and bounding box regression values predicted through the detection generator.

### 3.1. Collection and Pre-Processing of Road Driving Image Data

In this study, to configure an occluded crosswalk detection model, the self-driving dataset [23] collected from a self-driving vehicle known as IARA (Intelligent Robotic Autonomous Automobile) and the road driving image dataset [24] recorded using a GOPRO camera were used at the same time. IARA is a self-driving vehicle being developed at the LCAD of Universidade Federal do Espírito Santo. The involved vehicle consisted of a number of sensors. However, in this study, the learning images only consisted of the data collected from the camera for self-driving for image-based object detection. The IARA dataset included consecutive images consisting of general road driving images recorded during the daytime and road driving images recorded during the nighttime on weekdays. It consists of more than 35,000 images of day and night data and uses images with and without crosswalks. Among them, 12,748 images include crosswalks. The GOPRO dataset consisted of images recorded at 29.97 FPS using a GoPRO HERO 3 camera in Espírito Santo, Brazil. The involved images were divided into 29 sequences. Of the 29 sequences, 23 se-

quence images included the driving images of the vehicle passing through a crosswalk, and the remaining 6 sequence images consisted of the driving images of the vehicle not passing through a crosswalk. The dataset also consists of 11,070 daytime image data, of which 3964 images contain crosswalks. The two datasets are combined to use approximately 46,000 road images, of which 16,712 crosswalks are included. This study was aimed at designing a model capable of detecting the crosswalk objects to which the occlusion phenomenon is applied and not detecting objects that may be incorrectly detected as crosswalk objects at the same time. Accordingly, the sequence images that included a crosswalk were used as the training data of the detection model, and the sequence images that did not include a crosswalk were used as the data for hard-negative extraction and as the validation data. Figure 4 shows the pre-processing of road driving data for occlusion object detection.



**Figure 4.** Preprocessing road data for occlusion object detection.

As shown in Figure 4, input raw images represent a total of 35,625 frame-unit input raw images acquired from road driving images in diverse environments. The original resolution of the involved images is  $1920 \times 1080$  pixels. Since high-resolution images increases the complexity of the model algorithm for detecting the objects of interest and decreases the learning efficiency, in this study, the resolution was adjusted to  $640 \times 360$ . After the pixel pre-processing, the location of the ground truth bounding box of the crosswalk object was defined through the class labeling process of the objects of interest. This is an essential process for one-stage detector-based YOLO model training. In the final stage of pre-processing, the overall dataset is divided at a ratio of 8:2, and the training dataset for crosswalk occlusion detection and the test dataset were configured. Then, the training data were divided at a ratio of 3:1, and the training data for object detection neural network learning and the validation data used for evaluating the conformity of the model in training were configured.

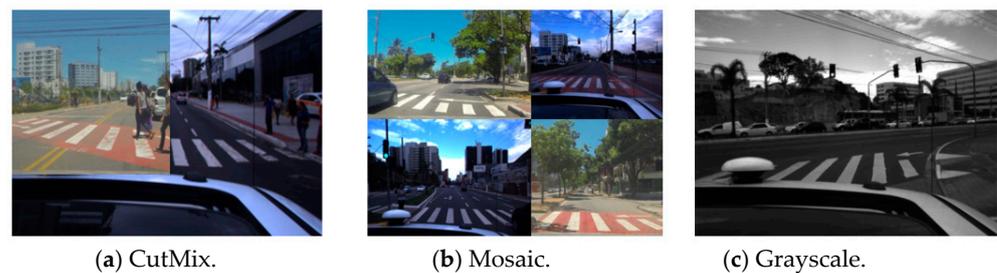
### 3.2. Augmentation Policy Optimization for Data Imbalance Problem Solving

The foreground-to-background imbalance problem that normally occurs in the process of detecting objects of interest in a road driving environment has a negative influence on the final detection performance [28]. This is caused by the intense imbalance between the number of positive examples serving as objects of interest and the number of negative examples serving as background objects, and causes overfitting. In this study, to prevent the data imbalance problem and overfitting, augmentation policies suitable for the learning model were strategically selected. In addition, the model pre-processing was optimized by adjusting the activation function and anchor box hyper parameters. Table 1 shows the augmentation policies applicable to learning for object detection, and also shows the detailed description of such policies [27]. Augmentation policy optimization is a process of strategically selecting policies that have a positive influence on the performance of learning data from among diverse policies such as MixUp, CutOut, CutMix, Mosaic, Blurring, Brightness, Contrast, Hue, and GrayScale and applying the selected policies to learning [29].

**Table 1.** Augmentation policies and description applicable to learning for object detection.

Index	Augmentation Policies	Detailed Description
1	MixUp	A policy that generates new data by randomly sampling and overlapping two input images.
2	CutOut	A policy that randomly truncates some regions of the input image so that they are hidden only in the first layer of the CNN.
3	CutMix	A policy that randomly samples two input images to create new data with a mixture of Ground Truth labels.
4	Mosaic	A policy that combines four random-input images into one training image at a certain rate.
5	Blurring	A policy that intentionally adds noise to the input image to blur the image.
6	Brightness	A policy that adjusts the brightness of the input image to a certain proportion to convert it to dark or bright.
7	Contrast	A policy that transforms contrast by adjusting contrast ratio of input images to a certain proportion.
8	Hue	A policy that transform hues by adjusting hue ratio of the input image to a certain proportion.
9	GrayScale	A policy that converts input images into black and white images to avoid considering color information.

This secures additional learning data required for training the deep learning model and prevents the data imbalance problem and overfitting [30]. In addition, diverse noise processings applied from the pre-existing data is applied to the model to acquire a more generalized model. A generalized model allows the detection of objects of interest in diverse driving environments [31]. However, indiscreetly applying the involved augmentation policies to learning has a negative influence on the performance of a detection model [32]. Since the optimized policies may also vary depending on the unique features of learning data, the user designing the detection model is required to make adequate adjustments [33]. Accordingly, in this study, images to which new augmentation policies are applied in sequence at intervals of 1000 iterations were added to the training data in the model learning process. This means that after 1000 training sessions, the weights for the learning model are extracted, adding images with new policies applied to the existing learning data to the model to continue learning into transfer learning. The policies added in phases went through the transfer learning of the pre-trained base YOLO model. In the transfer learning process for crosswalk occlusion detection, the bounding box regression values before and after applying particular policies were compared, and the residual was extracted. The residual was calculated based on the mAP used as the object detection performance index. When the gradient-based residual comparison is made between the location information of objects of interested predicted in the detection process and the location information of ground truth, the larger detection performance gradient means that the most recently applied augmentation policy satisfies the model optimality [34]. Accordingly, a comparison was made between the pre-existing detection mAP (mean Average Precision) value and the detection mAP value including the added policies, and the policies for enhancing the performance of occluded crosswalk object detection were strategically selected. A gradient-based performance residual comparison was made, and the increase/decrease in the mAP serving as the objective index was digitized to explore the optimal policy group. Figure 5 shows the result images of the training data to which augmentation policy optimization is applied according to the model learning process.



**Figure 5.** Result images of training data with augmentation policy optimization.

Figure 5a–c show the augmentation policies for enhancing the performance of occluded object detection that showed positive results when applied to the learning model in this study. Figure 5a CutMix is a policy that replaces a part of one image with a part of another image to newly configure the learning data, and shows amazing performance results for being an intuitive idea [35]. Figure 5b Mosaic is a policy that mixes and uses four learning images, and shows effective performance despite the small number of mini batches [36]. Figure 5c is a policy that processes the pre-existing color learning images into black-and-white images, and enables prompt image processing by simply delivering the luminosity information [37]. The MixUp, CutOut, Blurring, Brightness, Contrast, and Hue policies other than the three described above were not selected as the policies for model optimization, since the performance changes were reduced or insignificant in the transfer learning process. Accordingly, the CutMix, Mosaic, and Grayscale policies that showed detection performance enhancements through each policy performance evaluation were accepted and used in the final model learning. In addition, since the output value representing the detection performance varies depending on which activation function is used for model neural network configuration, it is very important to use activation functions suitable for learning [38]. In this study, a robust model suitable for normalization was developed and the Mish function that prevents overfitting was used as the activation function. This allows a bit of negative numbers to exist in the output value, and is particular in that the gradient is more smoothly displayed than that of the pre-existing ReLU series function. Since this is more advantageous in finding the global optimum value, it has a positive influence on the mean accuracy [38]. In addition, in the transfer learning process of the YOLO-based model, the k-means clustering method was applied to the bounding box of the ground truth included in the learning dataset, and the optimal anchor boxes suitable for crosswalk detection were found and used for learning. K-means clustering is an algorithm that aggregates given data into k clusters, minimizing the variance between each cluster and the distance difference. Generally, the method of using k-means clustering uses the euclidean method, but when applied to anchor boxes, it is difficult to find optimal anchor boxes because they rely too much on the center point. To improve this, we apply IoU (Intersection over Union) to extract anchor boxes. This had a positive influence on the crosswalk detection performance, since it took into consideration not just the pre-existing initially set value that simply took into consideration the aspect ratio according to pixels, but also the image features of the ground truth objects of learning data [39].

### 3.3. Detection Model of Occluded Object Based on YOLO Using Hard-Example Mining

After configuring the dataset to be used for model learning through the augmentation policy optimization, a custom learning model was conducted to detect the objects that occlude a crosswalk. To detect the occluded object on a road, it is necessary to extract the hard-positive data generated in the model learning process and include them as the input images of the learning model. An effective method to decrease the false-negative detection rate and false-positive detection rate of the objects of interest is to extract diverse hard-negative data that may occur in a road driving environment and have the model re-learn the data [40]. Hard-positive and hard-negative data are extracted from the hard example mining phase, Step-2 in Figure 3. The proposed process in this study extracts

data from the training datasets that are judged to be hard positive and negative and then reused for occluded crosswalks detection model training through appropriate pre-processing. Accordingly, in this study, in the YOLO-based model learning process for crosswalk detection, the hard-example data were extracted based on the confidence score threshold values and each boundary information of the detected classes. Then, the fine-tuning, to which transfer learning is applied, of the pre-trained model was performed. The convolution base of the pre-existing model serving as the part that extracts the features for detection from the input images was utilized to have just the classifier perform the re-learning process. In the road driving data for crosswalk detection, a hard-positive example is an example where the crosswalk object is not detected or is incorrectly detected as another object due to the occlusion phenomenon. A hard-negative example is an example where speed bumps and signs such as 'no speeding' and 'kids zone' for provision of traffic information other than crosswalks on roads are likely to be incorrectly detected as crosswalks. In the YOLO-based model learning process, the confidence score value is determined based on the IoU according to the pre-determined IoU (Intersection over Union) threshold value and on the object of interest existence probability, and the detection results are calculated according to the involved score value. In this study, the actual ground truth objects and hard-positive and hard-negative objects were classified through the involved confidence score information and bounding box information. Then, the extracted examples were added to the pre-existing learning data, the transfer learning was conducted, and the pre-trained model was renewed. As far as the model renewal is concerned, the feature vectors acquired from the locations predicted by the bounding box were set and learned as positive examples when they were included in the crosswalk objects, and the feature vectors acquired from the images not including crosswalk vectors were all set and learned as negative examples. In the model learning for occluded object detection, since the hard-negative feature vectors extracted from images not including objects of interest were all negative examples, they were included in the dataset for re-learning. The hard-negative feature vectors extracted from images including objects of interest were classified through the vector comparative calculation with the ground truth objects, and the filtered negative examples were included into the re-learned dataset. At this point, the bounding box information of the classes defined based on the extracted vectors were taken into consideration, the classifier was re-designed to classify the hard-negative examples as a separate class from the ground truth, and the involved examples were included into the new learning category. This induces the hard-negative objects likely to be detected as positive classes in the actual crosswalk detection process to be classified as classes of another category. Accordingly, this decreased the false-positive detection rate and enhanced the overall object detection performance. Lastly, since the hard-positive feature vectors that can be extracted from images including objects of interest were vectors included in or adjacent to the ground truth feature vectors, the bounding boxes of the involved images were reset by taking this into consideration, and then were included in the dataset for transfer learning. At this point, the IoU threshold value for hard-example estimation was set as a value below 0.5, and the feature vectors of the objects of interest were extracted. Algorithm 1 is an occluded object detection algorithm using hard-example mining.

As shown in Algorithm 1, the input value represents the learning images to which the CutMix, Mosaic and GrayScale policies acquired through augmentation policy optimization are applied. In the case where the objects of interest were included in the input image, when the IoU (Intersection over Union) threshold value of the predicted bounding box regression value and actual bounding box regression value was above 0.5, it was set as a positive example. Then, a hard-negative example was extracted through the predicted bbox regression value and confidence score within the positive example. In the case where the IoU threshold value of the predicted bounding box regression value and actual bounding box regression value was below 0.5, it was set as a negative example. Likewise, a hard-positive example was extracted through the predicted bbox regression value and confidence score within the negative example. Lastly, in the case where the objects of interest were

not included in the input image, since it meant that all the detected objects of interest were negative examples, they were deemed hard-negative examples and were extracted. The output value calculated through the hard-example mining algorithm represents the hard-negative examples and hard-positive examples extracted from the pre-existing positive examples and negative examples. Since this serves as a factor that increases the false-positive and false-negative detection rates in the process of detecting the objects of interest, the class bounding box of the involved data was reset and added to the pre-existing transfer learning model, and the data were re-learned.

---

**Algorithm 1.** Occluded Object Detection Algorithm using Hard-example Mining.

---

**Input:** training input images[n] applied cutmix, mosaic, and grayscale policies  
**Output:** Hard-negative\_example[n], Hard-positive\_example[n], Positive\_example[n], Negative\_example[n] Groups

```

confidence_score[n]←NULL
Positive_example[n]←NULL
Negative_example[n]←NULL
Hard-negative_example[n]←NULL
Hard-positive_example[n]←NULL
for  $n$  is 1 to number of training Input Images do
  if Ground Truth interest object is involved in Input Images[n]
    // IoU( Intersection over Union) = Overlapping Region/Combined Region
    if IoU of pred_bbox_reg( $n$ ) and GT_bbox_reg( $n$ )  $\geq$  0.5
      confidence_score[n]←Pr(object( $n$ )) * IoU
      if confidence_score[n]  $\geq$  0.5
        // Positive example mining with confidence score threshold
        Pos_feature_vector[n]←pred_bbox_reg( $n$ )
        Positive_example[n]←GT_class_labeling(Pos_feature_vector[n])
      else
        // Hard-negative mining with bbox comparison and reclassification
        Hard-Neg_feature_vector[n]←pred_bbox_reg( $n$ )
        bbox_tmp←Comp(Hard-Neg_feature_vector[n], GT_bbox_reg( $n$ ))
        filtering_tmp←Filtering(bbox_tmp)
        Hard-negative_example[n]←Class_redefine(filtering_tmp)
    else if IoU of pred_bbox_reg( $n$ ) and GT_bbox_reg( $n$ )  $<$  0.5
      confidence_score[n]←Pr(object( $n$ )) * IoU
      if confidence_score[n]  $<$  0.5 or confidence_score[n]  $\geq$  0.25
        // Hard-positive mining with class reclassification
        Hard-Pos_feature_vector[n]←pred_bbox_reg( $n$ )
        bbox_tmp←Class_redefine(Hard-Pos_feature_vector[n])
        Hard-positive_example[n]←bbox_tmp
      else // confidence_score[n]  $<$  0.25
        // Negative example mining with confidence score threshold
        Neg_feature_vector[n]←pred_bbox_reg( $n$ )
        Negative_example[n]←GT_class_labeling(Neg_feature_vector[n])
    else // Ground Truth interest object is not involved in Input Image
      if pred_bbox_reg( $n$ ) exists
        Hard-Neg_feature_vector[n]←pred_bbox_reg( $n$ )
        bbox_tmp←Hard-Neg_feature_vectors[n]
        Hard-negative_example[n]←Class_bbox_redefine(bbox_tmp)
  endfor
return

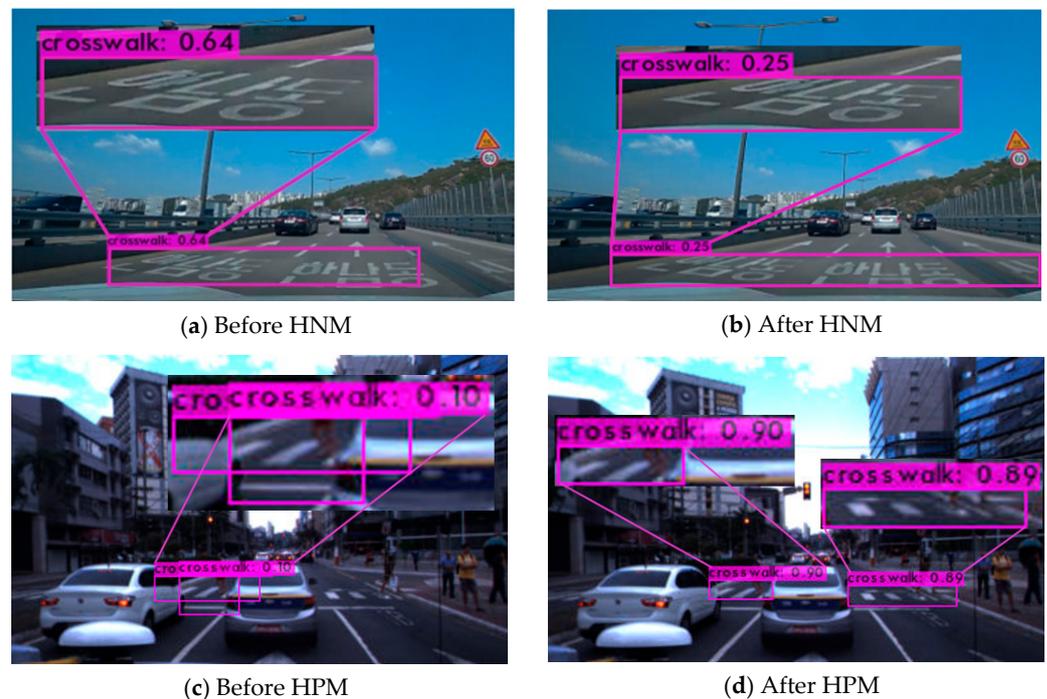
```

---

#### 4. Result and Performance Evaluation

As far as the test environment for implementing the proposed occluded object detection model is concerned, the hardware and operating system consisting of Windows10 Pro, AMD Ryzen 5 1600 Six-Core Processor, NVIDIA GeForce RTX 2080 and RAM 16GB

were used. In this study, a detection model of occluded object based on YOLO using hard-example mining and augmentation policy optimization was proposed. Figure 6 shows the crosswalk occlusion object detection results through the proposed hard-example mining.



**Figure 6.** Crosswalk Occlusion Object Detection Results with Hard-example Mining. HNM: Hard-negative Mining. HPM: Hard-positive Mining.

Figure 6a,b shows the detection model results acquired by using Seoul road driving images as the test data, and Figure 6c,d shows the detection model results acquired by utilizing the IARA data not used as learning data as the test data. Figure 6a shows the YOLO-based crosswalk detection results of a hard-negative example before the application of hard-negative mining. Based on the detection results, it was confirmed that the traffic information on a road that is not a crosswalk was incorrectly detected as a crosswalk, and that the confidence score value was 0.64 at that point. Figure 6b shows the detection results of a hard-negative example after the application of hard-negative mining, and it was confirmed that the confidence score value of the crosswalk was 0.25, and that the false-positive detection rate was greatly decreased. Figure 6c shows the crosswalk detection results of a hard-positive example before the application of hard-positive mining. The involved example was a representative occlusion phenomenon, and the confidence score values were 0.09 and 0.10, and it was unable to detect the crosswalk due to the block objects in front. However, as Figure 6d shows, based on the detection results acquired after the application of hard-positive mining, the confidence score values were enhanced to 0.90 and 0.89, and it was able to stably detect the objects that occluded the crosswalk in a road driving environment. The false-positive (64%) and false-negative (10%) objects generated in the pre-existing base model showed a false-positive detection rate of 25% and a true positive detection rate of 90%, respectively, meaning that the crosswalk object detection performance was greatly enhanced. As far as the real-life object detection is concerned, since the threshold value of the confidence score was 0.5, it was determined that the proposed model more stably detects crosswalk objects.

To evaluate the performance of the occluded object detection model proposed in this study, the performance evaluation for optimal augmentation policy configuration and the performance evaluation of the YOLO-based model to which the optimized augmentation policy and hard-example mining were applied were performed. To enhance the crosswalk

occlusion object detection performance during road driving, the performance changes in diverse augmentation policies were observed. The mAP@0.5 (mean Average Precision) was used as the performance index for evaluation, and the gradient value was used as the mAP@0.5-based quantitative index. mAP is a value that expresses the precision values calculated based on the actual object-detected recall changes as one index, and is a representative performance evaluation index of an object detection algorithm [41,42]. Equation (1) represents the gradient index.

$$\text{Gradient} = \left\{ \frac{\text{mAP after involving present policy}}{\text{mAP before involving present policy}} \times 100 \right\} - 100 \quad (1)$$

Equation (1) shows the relative changes in the mAP of the detection model before including the current augmentation policies and in the mAP of the detection model after including the current augmentation policies. At this point, the value is multiplied by 100 and 100 is subtracted from the value for the convenience of calculation through the normalization of performance changes. This allows each gradient index to be applied as a same-level scale, and has a positive effect on understanding the model performance changes. If the involved index value is a positive number, this means that the mAP increases, and, if the involved index value is a negative number, this means that the mAP decreases. Accordingly, the detection performance changes were objectively evaluated through the involved gradient index values. Table 2 shows the detection performance results for various augmentation policies [29,42].

**Table 2.** Detection performance results for various Augmentation policies. MU: MixUp. CO: CutOut. CM: CutMix. MO: Mosaic. BL: Blurring. BR: Brightness. CT: Contrast. HU: Hue. GS: Grayscale. Inv.mAP: mAP after involving present policy. Pre.mAP: mAP before involving present policy.

	MU	CO	CM	MO
Inv.mAP	0.7281	0.7355	0.7871	0.8360
Pre.mAP	0.7147	0.7281	0.7355	0.7871
Gradient	1.87	1.01	<b>7.01</b>	<b>6.21</b>
BL	BR	CT	HU	GS
0.8223	0.8380	0.8442	0.8392	0.8735
0.8360	0.8223	0.8380	0.8442	0.8392
−1.64	1.90	0.73	−0.6	<b>4.08</b>

As shown in Table 2, the CutMix policy showed the highest gradient value of 7.01. This showed better detection performance changes in comparison to the previously included MixUp and CutOut policies. CutMix is a policy created by mixing the strengths of the pre-existing MixUp and CutOut methodologies, and the objective performance difference was confirmed through the involved test results. The Mosaic policy showed the second highest gradient value of 6.21. Four random learning images were mixed into one image and were used for model learning, and it was confirmed that this policy had a positive influence on the learning efficiency and accuracy. Lastly, the Grayscale policy showed a high increase in terms of detection performance changes, and showed a gradient value of 4.08. The involved policy is a method that simply applies the black-and-white processing to the previously learned images. However, by examining the universal features of the learned images used in this study to detect crosswalk objects, it was possible to reason the grounds for high performance results. It was assumed that the difference between the black color of the asphalt-paved road and the white color displaying the crosswalk emphasized the features of the crosswalk through black-and-white processing and had a positive influence on the performance. Based on the test results, it was confirmed that the policies other than CutMix, Mosaic and Grayscale showed a mAP change that was either a negative number or low, so they were excluded from the optimal augmentation policy group. After completing the augmentation policy performance valuation, the optimal policies were selected and

applied to the YOLO model for detection together with hard-example mining. Then, the performance evaluation involving the following models was conducted: the BASE+BASE model serving as the pre-existing base model, the HEM+BASE model to which hard-example mining was applied without augmentation policy optimization, the APO+BASE model to which augmentation policy optimization was applied without hard-example mining, and this study's HEM+APO model to which augmentation policy optimization and hard-example mining were applied. Table 3 shows the detection performance results for the occluded object detection model proposed in this study and for the pre-existing model [8,18].

**Table 3.** Detection performance results for suggested model and existing base model.

	BASE+BASE	HEM+BASE	APO+BASE	HEM+APO (Ours)
mAP@0.5	75.14%	88.61%	87.95%	90.49%
F1-score	0.7502	0.8784	0.8568	0.9027

As shown in Table 3, a comparison was made between the BASE+BASE model and the HEM+BASE model, and it was confirmed that HEM+APO model showed the highest mAP value of 90.49%. In addition, as shown in the F1-score performance index, the proposed model showed an F1-score value of 0.9027 and showed higher performance than the other three models that showed F1-score values of 0.7502, 0.8784 and 0.8568, respectively. The HEM+BASE model showed an mAP value that was 13.47%p higher than that of the pre-existing base model. It seemed that the model to which hard-negative mining and hard-positive mining were applied showed better performance in detecting crosswalk occlusion objects compared to the pre-existing model, and the model demonstrated the effective true-positive detection of hard-example objects serving as causes that degrade detection performance. The APO+BASE model showed an mAP value that was 12.81%p higher than that of the pre-existing base model. It seemed that the model to which augmentation policy optimization were applied showed better performance compared to the pre-existing model. Thus, it demonstrated that more precise detection of occlusion objects can be achieved by applying CutMix, Mosaic, and Grayscale policies. In addition, the HEM+APO model showed an mAP value that was 1.88%p higher than that of the HEM+BASE mode. This model effectively overcame the data imbalance problem that is likely to occur in the object detection process for road driving through augmentation policy optimization.

In this study, we proposed a detection model of occluded objects based on YOLO using hard-example mining and augmentation policy optimization. In comparison to the pre-existing model, the proposed model more precisely detected the occluded objects during road driving, and the accuracy performance was enhanced as well. However, this study is limited in that the occlusion objects relating to multiple classes were not taken into consideration, because the proposed model was designed to detect occlusion objects relating to a single class known as a crosswalk. In addition, since the size of the IARA and GOPRO data sets for model training was not big enough, there is a possibility that the performance may be improved. In the future study, continuous tests must be conducted to take into consideration the occlusion phenomenon involving diverse objects of interest other than crosswalk objects that may occur in a road driving environment. In addition, diverse types of training data sets should be additionally collected and used for neural network learning and the pre-existing limitations should be supplemented to improve the performance of the occluded object detection model developed.

## 5. Conclusions

As limitations on the means of transportation are significantly decreased due to progress in road traffic technology, the number of modern people who own a vehicle continues to increase, and this leads to diverse traffic problems. As one of the plans to reduce the number of car accidents, self-driving technology development is attracting greater attention. The core of self-driving technology is object detection technology, and

studies on object detection during road driving are being actively conducted. However, since the several pre-existing studies did not carefully take into consideration the occlusion phenomenon that may occur in a road driving environment, they were limited in that they experienced an increase in the false-negative detection rate. Other previous surveys have used a variety of traditional and modern object detection algorithms to consider occlusion objects. However, the number of training data and the limitations for data pre-processing require further improvement. In addition, an increase in the false-positive detection rate that may occur due to objects or noises similar to the objects of interest has a negative influence on detection performance. In this study, a detection model of occluded object based on YOLO using hard-example mining and augmentation policy optimization was proposed. This model extracted the hard-positive objects resulting from the occlusion phenomenon in a road driving environment and the hard-negative objects likely to be incorrectly determined as objects of interest based on the confidence score and the boundary information of the detected classes. Then, fine-tuning that applies transfer learning to the pre-trained weights was conducted, and a robust detection model suitable for hard examples was developed. In addition, the model pre-processing process for occlusion detection and the augmentation policies were optimized through a gradient-based performance comparison, and the optimized policies suitable for learning data were strategically selected. Based on the evaluation results, the YOLO-based occluded object detection model proposed in this study showed an mAP value of 90.49% and an F1-score value of 90%. This showed the highest performance compared to the base model and the model to which no policy optimization was applied. Accordingly, as this model effectively overcame the data imbalance problem that degrades the performance of an object detection model in a self-driving environment and more precisely and more accurately detected the occluded objects that were difficult to detect in the past, it enabled object detection for self-driving in diverse environments.

**Author Contributions:** Conceptualization, S.-E.R. and methodology, K.-Y.C.; resources, S.-E.R. and K.-Y.C.; validation, S.-E.R. and K.-Y.C.; formal analysis, S.-E.R. and K.-Y.C.; investigation, S.-E.R. resources, S.-E.R.; data curation, S.-E.R.; writing—original draft preparation, S.-E.R.; writing—review and editing, S.-E.R. and K.-Y.C.; visualization, S.-E.R.; supervision, K.-Y.C.; project administration, K.-Y.C.; funding acquisition, K.-Y.C. Both authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the GRRRC program of Gyeonggi province. (GRRRC KGU 2020-B04, Image/Network-based Intellectual Information Manufacturing Service Research).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data sharing not applicable.

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

## References

1. Baek, J.W.; Chung, K. Pothole Classification Model Using Edge Detection in Road Image. *Appl. Sci.* **2020**, *10*, 6662. [CrossRef]
2. Traffic Accident Analysis System (TAAS). Available online: <http://taas.koroad.or.kr/> (accessed on 16 June 2021).
3. Namkoong, H.; Kang, S.; Won, Y.; Park, S. Identify research trends through big data analysis method for autonomous driving car. In *Proceedings of the Korea Technology Innovation Society Conference*; Asian Society for Innovation and Policy: Hong Kong, China, 2017; pp. 459–468.
4. National Highway Traffic Safety Administration (NHTSA). Available online: <https://www.nhtsa.gov/> (accessed on 16 June 2021).
5. Lian, J.; Yin, Y.; Li, L.; Wang, Z.; Zhou, Y. Small object detection in traffic scenes based on attention feature fusion. *Sensors* **2021**, *21*, 3031. [CrossRef]
6. Pegoraro, J.; Pflugfelder, R. The Problem of Fragmented Occlusion in Object Detection. *arXiv* **2020**, arXiv:2004.13076.
7. Huang, X.; He, P.; Rangarajan, A.; Ranka, S. Intelligent intersection: Two-stream convolutional networks for real-time near-accident detection in traffic video. *ACM Trans. Spat. Algorithms Syst. (TSAS)* **2020**, *6*, 1–28. [CrossRef]

8. Kim, M.J.; Kang, J.S.; Chung, K. Word-Embedding-Based Traffic Document Classification Model for Detecting Emerging Risks Using Sentiment Similarity Weight. *IEEE Access* **2020**, *8*, 183983–183994. [[CrossRef](#)]
9. Fan, B.B.; Yang, H. Multi-scale traffic sign detection model with attention. *Proc. Inst. Mech. Eng. Part D. J. Automob. Eng.* **2021**, *235*, 708–720. [[CrossRef](#)]
10. Lee, S.W.; Choi, B.; Kim, Y.S. A Vehicle License Plate Detection Scheme Using Spatial Attentions for Improving Detection Accuracy in Real-Road Situations. *J. Korea Soc. Comput. Inf.* **2021**, *26*, 93–101.
11. Ke, X.; Li, J. U-FPNDet: A one-shot traffic object detector based on U-shaped feature pyramid module. *IET Image Process.* **2021**. [[CrossRef](#)]
12. Ha, J.; Park, K.; Kim, M. A Development of Road Crack Detection System Using Deep Learning-based Segmentation and Object Detection. *J. Soc. E-Bus. Stud.* **2021**, *26*, 93–106.
13. Shin, D.H.; Chung, K.; Park, R.C. Prediction of Traffic Congestion Based on LSTM Through Correction of Missing Temporal and Spatial Data. *IEEE Access* **2020**, *8*, 150784–150796. [[CrossRef](#)]
14. Carranza-García, M.; Torres-Mateo, J.; Lara-Benítez, P.; García-Gutiérrez, J. On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data. *Remote Sens.* **2021**, *13*, 89. [[CrossRef](#)]
15. Zhou, X.; Koltun, V.; Krähenbühl, P. Probabilistic two-stage detection. *Arxiv Prepr.* **2021**, arXiv:2103.07461.
16. Gai, R.; Chen, N.; Yuan, H. A detection algorithm for cherry fruits based on the improved YOLO-v4 model. *Neural Comput. Appl.* **2021**, 1–12.
17. Yu, J.; Zhang, W. Face mask wearing detection algorithm based on improved YOLO-v4. *Sensors* **2021**, *21*, 3263. [[CrossRef](#)]
18. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
19. Jeon, S.; Kim, D.; Jung, H. YOLO-based lane detection system. *J. Korea Inst. Inf. Commun. Eng.* **2021**, *25*, 464–470.
20. Du, H.; Zhu, G.; Zheng, J. Why travelers trust and accept self-driving cars: An empirical study. *Travel Behav. Soc.* **2021**, *22*, 1–9. [[CrossRef](#)]
21. Ibrahim, H.; Salem, A.D.A.; Kang, H.S. Real-Time Weakly Supervised Object Detection Using Center-of-Features Localization. *IEEE Access* **2021**, *9*, 38742–38756. [[CrossRef](#)]
22. Aslani, S.; Mahdavi-Nasab, H. Optical flow based moving object detection and tracking for traffic surveillance. *Int. J. Electr. Comput. Energetic Electron. Commun. Eng.* **2013**, *7*, 1252–1256.
23. Possatti, L.C.; Guidolini, R.; Cardoso, V.B.; Berriel, R.F.; Paixão, T.M.; Badue, C.; Oliveira-Santos, T. Traffic light recognition using deep learning and prior maps for autonomous cars. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; IEEE: New York, NY, USA, 2019; pp. 1–8.
24. Berriel, R.F.; Rossi, F.S.; de Souza, A.F.; Oliveira-Santos, T. Automatic large-scale data acquisition via crowdsourcing for crosswalk classification: A deep learning approach. *Comput. Graph.* **2017**, *68*, 32–42. [[CrossRef](#)]
25. Ning, C.; Menglu, L.; Hao, Y.; Xueping, S.; Yunhong, L. Survey of pedestrian detection with occlusion. *Complex Intell. Syst.* **2021**, *7*, 577–587. [[CrossRef](#)]
26. Kulkarni, R.; Dhavalikar, S.; Bangar, S. Traffic light detection and recognition for self driving cars using deep learning. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 16–18 August 2018; IEEE: New York, NY, USA, 2018; pp. 1–4.
27. Xie, H.; Wu, Z. A Robust Fabric Defect Detection Method Based on Improved RefineDet. *Sensors* **2020**, *20*, 4260. [[CrossRef](#)] [[PubMed](#)]
28. Ryu, S.E.; Kim, H.J.; Koo, B.K.; Kwon, H.J.; Park, R.C.; Chung, K. Data Bias Optimization based Association Reasoning Model for Road Risk Detection. *J. Korea Converg. Soc.* **2020**, *11*, 1–6.
29. Hataya, R.; Zdenek, J.; Yoshizoe, K.; Nakayama, H. Meta Approach to Data Augmentation Optimization. *arXiv* **2020**, arXiv:2006.07965.
30. Yoo, H.; Han, S.; Chung, K. Diagnosis Support Model of Cardiomegaly based on CNN using Resnet and Explainable Feature Map. *IEEE Access* **2021**, *9*, 55802–55813. [[CrossRef](#)]
31. Ryu, S.E.; Kwon, H.; Kim, M.J.; Chung, K. Deep Autoencoder Model for Unsupervised Anomaly Detection in Traffic Safety. In Proceedings of the International Conference on Convergence Content, Jeju-si, Korea, 21–23 October 2020; pp. 53–54.
32. Luo, C.; Zhu, Y.; Jin, L.; Wang, Y. Learn to augment: Joint data augmentation and network optimization for text recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 13746–13755.
33. Zoph, B.; Cubuk, E.D.; Ghiasi, G.; Lin, T.Y.; Shlens, J.; Le, Q.V. Learning data augmentation strategies for object detection. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020; pp. 566–583.
34. Kim, J.C.; Chung, K. Hybrid Multi-modal Deep Learning using Collaborative Concat Layer in Health Bigdata. *IEEE Access* **2020**, *8*, 192469–192480. [[CrossRef](#)]
35. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 6023–6032.
36. Hao, W.; Zhili, S. Improved Mosaic: Algorithms for more Complex Images. *J. Phys. Conf. Ser.* **2020**, *1684*, 012094. [[CrossRef](#)]
37. Gong, Y.; Zeng, Z. An Effective Data Augmentation for Person Re-identification. *arXiv* **2021**, arXiv:2101.08533.

38. Ryu, S.E.; Shin, D.H.; Chung, K. Prediction model of dementia risk based on XGBoost using derived variable extraction and hyper parameter optimization. *IEEE Access* **2020**, *8*, 177708–177720. [[CrossRef](#)]
39. Kim, H.J.; Baek, J.W.; Chung, K. Associative Knowledge Graph using Fuzzy Clustering and Min-Max Normalization in Video Contents. *IEEE Access* **2021**, *9*, 74802–74816. [[CrossRef](#)]
40. Chen, K.; Chen, Y.; Han, C.; Sang, N.; Gao, C. Hard sample mining makes person re-identification more efficient and accurate. *Neurocomputing* **2020**, *382*, 259–267. [[CrossRef](#)]
41. Henderson, P.; Ferrari, V. End-to-end training of object class detectors for mean average precision. In *Asian Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 198–213.
42. Baek, J.W.; Chung, K. Multi-Level Health Knowledge Mining Process in P2P Edge Network. *IEEE Access* **2021**, *9*, 61623–61634. [[CrossRef](#)]