Research Article

Research on Subway Pedestrian Detection Algorithm Based on Big Data Cleaning Technology

Zhuoyang Lyu

College of Science, Purdue University, USA

Correspondence should be addressed to Zhuoyang Lyu; lyurobin914@gmail.com

Received 6 August 2021; Revised 13 October 2021; Accepted 20 October 2021; Published 7 December 2021

Academic Editor: Rajesh Kaluri

Copyright © 2021 Zhuoyang Lyu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The pedestrian detection model has a high requirement on the quality of the dataset. Concerning this problem, this paper uses data cleaning technology to improve the quality of the dataset, so as to improve the performance of the pedestrian detection model. The dataset used in this paper is obtained from subway stations in Beijing and Nanjing. The data images' quality is subject to motion blur, uneven illumination, and other noisy factors. Therefore, data cleaning is very important for this paper. The data cleaning process in this paper is divided into two parts: detection and correction. First, the whole dataset goes through blur detection, and the severely blurred images are filtered as the difficult samples. Then, the image is sent to DeblurGAN for deblur processing. 2D gamma function adaptive illumination correction algorithm is used to correct the subway pedestrian image. Then, the processed data is sent to the pedestrian detection model. Under different data cleaning datasets, through the analysis of the detection results, it is proved that the data cleaning process significantly improves the detection model's performance.

1. Introduction

Researches regarding data cleaning are first appeared in the United States, on the correction of social security number errors. The early research on data cleaning mainly focused on information data. The main research contents are as follows: (1) detection and elimination of abnormal data; (2) detection and elimination of approximate duplicate data; (3) data integration; (4) domain specific data cleaning.

Big data is the symbolic representation of this information-driven world. It has four characteristics: volume, variety, value, and velocity. It is gradually independent of software products and even dominated the development of some software products, such as Hadoop, Oracle, Hive, and Spark. Today, people can obtain massive amount of data from a variety of ways. After obtaining data, we often need to process them differently according to our specific purpose and extract valuable information from them. In order to get valuable information to meet people's needs, the data obtained should be reliable and accurate in reflecting the actual situation. However, the first-hand data

we are able to collect is often dirty. Dirty data refers to inconsistent, inaccurate data resulting from human errors. Dirty data itself has the characteristics of inconsistency and inaccuracy, which directly affect its explicit and implicit value, that is, directly affect its quality [1].

The steps of data cleaning can be divided into the following steps:

- (1) Demand analysis. The purpose of this stage is to clarify the format of effective data by analyzing the application field and application environment of the data and then the goal of data cleaning [2]
- (2) Preprocessing. Through data analysis technology, we identify the quality problems existing in the dataset and summarize information regarding data's quality
- (3) Determination of cleaning rules. This part analyzes the root causes of noise to define data cleaning rules. Different datasets have different characteristics, so the rules need to be selected to suit specific dataset [3]

- (4) Cleaning and correction. This part involves cleaning the data according to the defined cleaning rules, using related technologies to correct the dirty data, and meeting the requirements of demand analysis. There are two general divisions among common data cleaning methods: repeated data detection and outlier data detection [4]. Repeated data detection includes field-based detection algorithm Levenshtein distance algorithm [5] and cosine similarity function algorithm. Levenshtein distance algorithm is easy to implement. Cosine similarity algorithm is more used to detect text similarity. The smaller the value of similarity measure obtained by this algorithm, the more similar the individuals are. Record-based detection algorithms include N-grams algorithm, clustering algorithm, SNM algorithm, and MPN algorithm [6]. N-grams algorithm generates a hash table and then judge the similarity between records according to the hash table; clustering algorithm classifies similar data into one class through calculation. The implementation of SNM algorithm is relatively easy, but it depends on keywords to a large extent and has strong dependency. The advantage of MPN algorithm is that it can collect the repeated data more comprehensively, but it is more cumbersome to use. Outlier detection is used to detect objects that are significantly different from other data points—outliers. Outlier detection algorithms mainly include the algorithm based on aggregate model, the algorithm based on proximity, the algorithm based on density, and the algorithm based on clustering [7]. The detection steps based on the statistical model algorithm are as follows: first, the data model is established, then the detection algorithm conducts analysis according to the model to identify outliers. Proximity-based algorithms define the proximity between objects. The core of density-based algorithm is to detect the local density of an object. When its local density is lower than that of most objects in the neighborhood, it is judged as an outlier. Cluster-based algorithms are used to find groups of objects that are locally strongly related, while outliers are objects that are not strongly related to other objects. After the test is completed, correct the wrong data according to the test results to achieve the purpose of cleaning.
- (5) Verification. Finally, the corresponding calibration operation is used to verify whether the cleaned data meets the requirements. If it does not meet the task requirements, the cleaning rules needs to be modified, the data cleaning process should be repeated, and the results can be verified and evaluated again. R-CNN [8] (region-based convolutional neural networks) algorithm, which was proposed in 2013, is a region-based CNN, which can be applied to the industrial field. Later, the region-based CNN has been further optimized, resulting in many better performance region-based convolutional neural networks, for example, the current mainstream detector: faster R-CNN [9]. The detector based on

deep learning learns the features of the target autonomously through the backbone network in the training process while the traditional algorithm needs manually set features. The method based on deep learning is more robust and is easier to generalize

With a large number of scholars dedicated to this field, the algorithm is being improved continuously at present. On the contrary, the performance of the model is limited at the data level. Right now, the data quality of pedestrian datasets KITTI [10], Caltech [11], and CityPersons [12] published is relatively general, which means it is usually affected by uneven illumination and motion blur, the two prominent problems.

This paper designs the following steps through the data cleaning of the collected mass subway pedestrian pictures.

- (1) Demand analysis. In view of these two prominent problems, this paper collects, cleans, and makes a dataset of subway pedestrians from real life scenes. Aiming at the image quality requirements of subway pedestrian detection task, we produce a high-quality subway pedestrian dataset
- (2) Preprocessing. In the preprocessing step, the variance of the image is calculated according to the Laplace operator, the degree of blur of the image is identified, and the distribution of fuzzy image and clear image in the collected image is statistically analyzed
- (3) Set cleaning rules. In this paper, a threshold is set according to the preprocessing results. If the variance of the image is less than the threshold, it will be regarded as a fuzzy image and its data will be cleaned
- (4) Cleaning and calibration. For blurred images, this paper uses a DeBlurgan network for deblurring; for images with uneven illumination distribution, the illumination intensity is adjusted adaptively by using two-dimensional gamma function
- (5) Check. In this paper, the dataset obtained by using different data cleaning rules will be sent into the classical YOLOV3 network to test the performance of the model and analyze the effectiveness of the data cleaning method used in this paper in the pedestrian detection task

The structure of this paper is as follows: the second section is the introduction and quality analysis of the dataset. The third section is the method of data cleaning and verification we used. The fourth section is the experimental design and results, and the fifth section is the conclusion of this paper.

2. Dataset

2.1. Subway Pedestrian Dataset. Due to the relatively dense number of passengers in the subway station and the height

and angle of the monitoring camera, when the crowd is dense, pedestrian's trunk is easy to block each other, and the head-shoulder positions are generally relatively complete. Therefore, the detection model based on the head-shoulder positions is established for the detection of pedestrians. Subway pedestrian dataset was collected by monitoring video of Beijing subway station. First, the video data was read in frame by frame, and the generated pictures were stored locally in JPG format. With reference to the format of VOC2007 dataset, a total of 17774 original pictures of multiple scenes were processed and annotated.

Our dataset is a pedestrian dataset obtained from subway station, which contains a large number of occlusion scenes. It can effectively evaluate the robustness of the detector to occlusion problems. It contains a total of 9,000 images in training set. These pictures are all from some subway stations in Beijing and Nanjing. The average number of pedestrians per picture is 13.36, more than Caltech and CityPersons. As shown in Table 1, our dataset is more challenging than the Caltech and CityPersons benchmark datasets.

Subway pedestrian dataset was made, and labelme software was used to mark the pedestrian head-shoulder positions with a rectangular frame. The marking box should contain as much pedestrian head and shoulder positions as possible while containing as little background information as possible. The obtained subway pedestrian dataset contains the passenger flow situation at different times and places in the subway station. When annotated, XML files are generated in the same folder; as shown in Figure 1, the labelme software and the information are contained in the XML file.

2.2. Quality Analysis of Dataset. The collected subway pedestrian images are inevitably affected by various factors in the process of acquisition, storage, and transmission. And then produce different types of distortion and different degrees of distortion, in which blur distortion is the most common. Blur distortion leads to the degradation of image quality, which affects the accuracy of pedestrian detection. So, we use the method of blur detection to analyze the image quality of the dataset.

This paper uses Laplacian function of OpenCV to detect image blur. Because Laplacian operator is used to measure the second derivative of the image, it can emphasize the region with fast changing density in the image, that is, the boundary region. In the general picture, the boundary is clear, so the variance will be larger; however, there is little boundary information in the blurred image, so the variance will be small. Firstly, a channel of the image is selected and convoluted with 3×3 convolution kernel to calculate the variance of the output. The formula of convolution is shown as

$$\int_{-\infty}^{+\infty} f(t)y(x-t)dt. \quad (1)$$

$f(t)$ represents the convolution kernel, and $y(x-t)$ represents a channel of the image.

If the variance is less than the set threshold, it is regarded as a blurred image. We set the threshold value to 40. If the function return value is less than 40, it is considered as a blurred image. If the function value exceeds 40, it is a clear image. At the same time, we divide the blurred image into different grades. The range of 0-10 is regarded as level 1 blur; 10-20 is two-level blur; 20-30 is grade 3, and 30-40 is grade 4.

We analyze the image quality of all the datasets. The images with variance greater than 40 are regarded as clear images, and the images with variance less than 40 are processed for subsequent deblurring.

As shown in Figure 2, there are four levels of blur distribution in the subway pedestrian dataset. It can be seen from the observation that a large number of images are gathered in the first and second levels of blur, so it is necessary to deblur the dataset. More than 60% of the images in subway pedestrian dataset are blurred. A large number of blur samples will affect the training effect of the network, so it is necessary to deblur the images in the dataset.

3. Data Cleaning Algorithm

3.1. DeblurGAN. Motion blur image is generated by relative movement of equipment and target during image acquisition process. In subway station, the monitoring camera is generally on the high place with a certain inclination angle. When making datasets by intercepting monitoring video frames, the blurred picture will appear when passengers move quickly, which not only affects the image quality, but also makes it difficult to detect pedestrians.

DeblurGAN, an end-to-end learning method for generating network and content loss based on conditional antagonism, removes blurring of images due to pedestrian movements. When the image to be detected is blurred, first, blurring the image can improve the accuracy of pedestrian detection.

The blurring of the image can be seen as the convolution of the original image and the convolution kernel plus additive noise. It can be expressed as

$$I_B = k * I_S + N. \quad (2)$$

I_B and I_S represent blurred image and clear image, respectively; k is unknown blur kernel; N is additive noise. Most algorithms rely on the classic Lucy Richardson algorithm [13] and Wiener or Tikhonov filter to perform deconvolution operation and obtain the estimation [14], which is to restore the blurred image with known blur kernel. But usually, the blur function is unknown, so it is uncertain to find the blur function for each pixel. DeblurGAN processes the blurred image I_B as input without information about the blur kernel to get a clear image I_S . In DeblurGAN training phase, CNN is trained as generator network G_θ and discriminator network D_θ by constructing Generative Adversarial Network, with pairs of blur image and clear image as input, and finally, clear image is reconstructed through the means of adversarial. After the training, the whole process of deblurring is completed by the trained

TABLE 1: Our dataset compare with public benchmark datasets.

	Caltech	KITTI	CityPersons	COCOPersons	Our datasets
Images	42,782	3,172	2,975	64,115	9,000
Persons	13,674	2,322	19,238	257,252	120,325
Person/image	0.32	0.63	6.47	4.01	13.36

FIGURE 1: labelme software.

FIGURE 2: Image blur level distribution in dataset.

network G_θ . In this case, only the input blurred image I_B is restored to get a clear image I_S , so as to achieve the result of motion blur removal.

DeblurGAN's loss function consists of two parts: antiloss L_{GAN} and content loss L_x . It can be expressed as

$$L = L_{GAN} + \alpha L_x. \quad (3)$$

α is 100 in this deblurring experiment, which means the weight of content loss. WGAN-GP is used as the antiloss function, which is robust to the training of the generated

network. It can be expressed as

$$L_{GAN} = \sum_{n=1}^N -D_\theta(G_\theta(I_B)). \quad (4)$$

Content loss uses the perceptual loss function, which is a simple L2 loss. The difference of each layer's feature map between the generated image CNN and the target image CNN is calculated, and the final cumulative error is the perceptual loss. The calculation in Equation (5) shows that $\phi_{i,j}$ is the feature map obtained by the i activated convolution layer before the j largest pooling layer of vgg19 network trained on ImageNet dataset; $W_{i,j}H_{i,j}$ represents the dimension of the feature map.

$$L_x = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^W \sum_{y=1}^H \left(\phi_{i,j}(I_S)_{x,y} - \phi_{i,j}(G_\theta(I_B))_{x,y} \right)^2. \quad (5)$$

3.2. 2D Gamma Function. In subway stations, pedestrians block each other, and different areas have different light irradiation intensity, which often leads to uneven illumination around pedestrians. This is mainly reflected in the insufficient illumination in some areas of the image, and the excessive illumination in some areas of the image. Some image details cannot be extracted in the test, which seriously affects the pedestrian detection results. Therefore, it is necessary to correct the uneven illumination of subway pedestrian image to eliminate the influence caused by uneven illumination as far as possible.

Generally speaking, the digital image can be regarded as a 2D function $f(x, y)$, which is obtained by multiplying the incident light component $i(x, y)$ and the object surface reflection component $r(x, y)$,

$$f(x, y) = i(x, y) * r(x, y). \quad (6)$$

The spatial relationship is shown in Figure 3. For images with uneven illumination, it is the uneven distribution of incident illumination component that causes the image brightness value to be too large in areas with strong illumination, while the image brightness value in areas with weak illumination is too small. It is very important to extract the incident light component from the illumination correction of the image with uneven illumination. The illumination component is extracted by multiscale Gaussian function and the Gaussian function formula as shown

FIGURE 3: Spatial relation of object illumination.

in Equation (7).

$$G(x, y) = \lambda \exp\left(-\frac{x^2 + y^2}{c^2}\right), \quad (7)$$

where c is the scale factor and λ is the normalization constant, and the Gaussian function is required to meet the normalization condition $\iint G(x, y) dx dy = 1$. The Gaussian function is convolved with the input image $F(x, y)$ to obtain the estimated value of the illumination component $I(x, y) = F(x, y)G(x, y)$. The multiscale Gaussian function method is adopted to extract the illumination component by using the Gaussian function of different scales, and then, the illumination component is weighted. Finally, the estimated value of the illumination component is obtained. The formula is shown in Equation (8), which ω_i represents the weight coefficient of the illumination component corresponding to the Gaussian function of the i scale.

$$I(x, y) = \sum_{i=1}^N \omega_i [F(x, y)G_i(x, y)]. \quad (8)$$

After the illumination component is extracted, an adaptive brightness correction method based on the 2D gamma function is constructed. According to the distribution characteristics of the illumination component, the parameters of the 2D gamma function are adjusted adaptively, and the image with uneven illumination is corrected, so as to reduce the brightness value of the area with too strong illumination and increase the brightness value of the area with too low illumination, so as to achieve the effect of processing the image with uneven illumination. This allows the model to learn more details about the dark parts of the image. For the input image $F(x, y)$, assuming that the extracted illumination component is $I(x, y)$, the improved 2D gamma function expression is shown in Equation (9), which $O(x, y)$ represents the brightness value of the corrected image, γ represents the index value of brightness enhancement, and m represents the mean brightness value of the illumination component.

$$O(x, y) = 255 \left(\frac{F(x, y)}{255} \right)^\gamma, \quad \gamma = \left(\frac{1}{2} \right)^{(m - I(x, y)) / m}. \quad (9)$$

3.3. Pedestrian Detection Based on YOLOV3. Object detection algorithms based on deep learning mainly include two types, one based on anchor frame and divided into two stages and one stage. Two-stage detection methods, such as RCNN series et al. [15, 16], first generate a group of candidate bounding boxes that may contain targets by using the region proposal module and then classify and regression these borders by using deep convolutional neural network [17, 18]. One-stage detection methods, such as YOLO series [19, 20] and SSD [21], unify all modules of target detection into a single convolutional network, enabling it to simultaneously predict the probability of multiple bounding boxes and categories. The other is anchor-free detection method, such as CornerNet [22] and ExtremeNet [23]. As a one-stage object detection method, YOLOV3 can locate the object in the input image and predict its category at the same time, thus transforming the object detection problem into a regression problem. The overall detection process of its network is shown in Figure 4.

We use the PyTorch framework, and the resolution of the input image is $416 * 416$. After passing through multiple convolution layers, the data of three scales will be output. If we use the COCO dataset, there are 80 categories, namely, (N,255,13,13), (N,255,26,26), and (N,255,52,52). Since there is only one type of target to be detected in subway pedestrian detection process (marked with head-shoulder), the number of output categories of YOLOV3 network is 1 by modifying the length of network prediction tensor is 18, and the three scales are (N,18,13,13), (N,18,26,26), and (N,18,52,52), respectively. Each figure is divided into 3 priori box positions on the grid of 13, 13, 26, 26, 52, and 52.

4. Experimental Results and Analysis

4.1. DeBlurGAN Removes Blur. The entire structure of the DeBlurGAN training network for motion blur removal is shown in Figure 5, where the generator network takes the blur image as input and produces the reconstructed image. During training, the discriminant network takes the reconstructed image and the original clear image as input and estimates the distance between them. The generator network structure, shown in Figure 6, consists of two step convolution blocks with one half of the stride size, nine residual blocks (ResBlocks), and two transpose convolution blocks. Each ResBlock consists of a convolution layer, an instance normalization layer, and a ReLU activation layer. Add a missing regularization with a probability of half after the first convolution layer in each ResBlock. In addition, there is a global skip connection called ResOut. The DeBlurGAN discriminator network architecture still uses Patch-GAN from Pix2Pix. In this paper, through the use of GoPro dataset (part), a total of 1146 pairs of $720 * 720$ blur-clear image pairs were taken from different scenes, 200 iteration training was carried out in the TensorFlow framework of Linux system, and the training result model was saved every 20 times by modifying the network settings. For the blur image of subway pedestrians, there is no image processing and no corresponding clear image, so the supervised method cannot

FIGURE 4: Detection process of YOLOv3.

FIGURE 5: The architecture of DeblurGAN training network.

FIGURE 6: The generator architecture of DeblurGAN.

be used to conduct deblurring training on this dataset. However, the dataset is derived from the actual subway scene that needs to be deblurred and has practical application significance, so it can be used as a test dataset. By calling the training model of subway pedestrian to deal with the blur dataset, to the whole process of blur network of training alone, because the original network output picture image resolution is relatively low, the changes to the network are not reducing image characteristics to the original size to save to the pedestrian subway after blur images.

DeblurGAN was used to process the subway pedestrian dataset to obtain the deblurred image (as shown in Figure 7). It can be seen that compared with the original Figure 7(a), the deblurred image in Figure 7(b) is clearer, the detailed texture in the image is more prominent, and the pedestrian contour on the left of the image is more obvious. It is convenient to detect the head-shoulder of pedestrians in the image. In Figure 8, the model obtained from image training before and after deblurring is detected through the pedestrian detection network. As shown in Figure 8(a), the two pedestrians at the bottom of the image are not detected. As shown in Figure 8(b), the texture details

are more visible in the deblurred image, so they are successfully detected.

4.2. Adaptive Luminance Correction Algorithm for Two-Dimensional Gamma Function. Using multiscale Gaussian function to extract the subway dataset nonuniform illumination image of light weight, structure based on 2D adaptive brightness adjustment function of the Gamma function, and using the distribution characteristics of light weight adaptively adjust the 2D gamma function parameter and adaptive correction in nonuniform illumination image processing. On the premise of effectively retaining the effective information of the original image, the purpose of correcting the image with uneven illumination can not only effectively improve the visual effect of the pedestrian detection image but also find more details of the dark place in the image. The RGB color space of the input pedestrian detection image is transferred to the HSV space, and the V (brightness) component of the HSV space is operated without affecting the color information of the image. The multiscale Gaussian filter of Retinex is used to obtain the incident light component, and then, the 2D gamma function is used. The image

FIGURE 7: Comparison between the images before and after deblurring.

FIGURE 8: Comparison between the detection images before and after deblurring.

FIGURE 9: Flow chart of nonuniform illumination correction algorithm.

brightness is corrected by changing the brightness, and then, the image is synthesized with T(tonal) and S(saturation) components, and then, the image is returned to the RGB color space to output the corrected image of uniform illumination. In this paper, the illumination correction program was written by MATLAB under Windows to deal with the image dataset of subway pedestrians with uneven illumination in batches. In order not to affect the subsequent entry into the object detection network, the illumination correction pictures were saved in full size. The algorithm flow chart is shown in Figure 9.

The illumination component is extracted from Figure 10(a) of subway pedestrians with uneven illumination to obtain the Figure 10(b) of the corresponding light component. As shown in Figure 10(a), the brightness of the middle

part of the original image is larger due to the illumination of subway lights, while the brightness is darker if there is no direct illumination around. The middle part of the Figure 10(b) after the illumination component is also larger. Figure 10(c) of illumination correction processing was obtained by self-adaptive correction processing. Compared with the original image, the brightness of the middle part decreased, and the brightness of the four corners increased significantly.

After testing the model obtained from the training of the image dataset before and after the illumination correction treatment, the comparison of the detection images before and after the illumination correction treatment is shown in Figure 11. Figures 11(a) and 11(c) are the preillumination models to detect the images before illumination processing,

FIGURE 10: Comparison between the images before and after illumination correction.

FIGURE 11: Comparison between the detection images before and after illumination correction.

TABLE 2: The default anchor box size before and after K -means clustering.

Feature maps of different sizes	13×13	26×26	52×52
Anchor before clustering	(116, 90)	(30, 61)	(10, 13)
	(156, 198)	(62, 45)	(16, 30)
	(373, 326)	(59, 119)	(33, 23)
Anchor after clustering	(135, 120)	(99, 92)	(48, 48)
	(159, 166)	(105, 113)	(63, 71)
	(202, 230)	(119, 146)	(80, 83)

TABLE 3: Datasets and models corresponding to different image processing.

Pedestrian image processing	Dataset	Model
The original image	Dataset I	Model I
Deblurring	Dataset II	Model II
Illumination correction	Dataset III	Model III
Deblurring+ illumination correction	Dataset IV	Model IV
Illumination correction + deblurring	Dataset V	Model V

TABLE 4: Comparison of test results of corresponding models in each dataset.

Model	Number of test-set images	Number of pedestrian instances	Recall	Precision	mAP
Model I	3555	30717	0.83	0.68	0.78
Model II	3555	30717	0.85	0.69	0.79
Model III	3555	30717	0.85	0.83	0.83
Model IV	3555	30717	0.87	0.81	0.84
Model V	3555	30717	0.88	0.78	0.85

and it is found that there are false detection and redundant detection frames, etc. After illumination correction, the brightness of pedestrians in the dark environment around the picture will increase. Figures 11(b) and 11(d) accurately detect pedestrians without false detection and redundant detection frames.

4.3. Pedestrian Detection Based on YOLOv3. In this paper, Yolov3 network is used to train and detect the subway pedestrian dataset. Three anchor frames are set at each scale. Before training, K -means clustering is performed on the label frame of the subway pedestrian dataset in this paper to calculate the initial value of the anchor frame in the training set, making the size of the anchor frame more consistent with the size of the pedestrian head and shoulder. The size of the default Anchor box before and after clustering is shown in Table 2.

The original dataset of subway pedestrians is named dataset I.

Dataset II was obtained from dataset I after DeblurGAN deblurring.

Dataset III was obtained from dataset I after illumination correction.

Dataset IV was obtained from dataset I after DeblurGAN deblurring and illumination correction.

Dataset V was obtained from dataset I after illumination correction and DeblurGAN deblur processing.

As shown in Table 3, the deep convolutional neural network YOLOV3 was used for multiple rounds of training under the framework of PyTorch. The detection models obtained from the corresponding training of five datasets were named as model I, model II, model III, model IV, and model V, respectively.

The same YOLOV3 detection network under the PyTorch framework was used to test the models I, II, III, IV, and V obtained by training, respectively. The model file sizes of the five models were almost the same. When the speed was tested, 10 of the targets in the video had speeds of around 17-19 fps below them, and 10 of the targets had speeds of around 13-16 fps above them. The number of test pictures is 3555, including 30,717 subway pedestrian head-shoulder targets. The model detection results are shown in Table 4. It can be seen that image DeblurGAN deblurring, uneven illumination adaptive correction, first DeblurGAN deblurring followed by uneven illumination adaptive correction, first DeblurGAN deblurring followed by uneven illumination adaptive correction, and then DeblurGAN deblurring will all improve the mean detection accuracy (mAP) of the model. Among them, model IV and model V are higher than model II and model III in mAP, indicating that

the combined operation effect of two treatment methods of DeblurGAN deblurring and illumination uneven adaptive correction is better than that of one treatment without any sequence.

5. Conclusion

In this paper, it is considered that the metro pedestrian dataset with large data volume and low data quality is the main reason for the poor performance of pedestrian detection model. Therefore, the data cleaning technology is introduced into the subway pedestrian detection system. We first use Laplace operator to carry out blur detection on subway pedestrian images and divide the images in the dataset into clear pictures and blur pictures. We also used the DeblurGAN network to deblur the blurred image and further used the 2D gamma function to equalize the light in the image. Through the use of different combination of data cleaning methods and the verification of YOLOV3 algorithm, the rationality of our hypothesis is verified, and the performance of pedestrian detection algorithm is significantly improved by data cleaning.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no competing interest.

References

- [1] F. Wenfei, "Extending dependencies with conditions for data cleaning," in *8th IEEE International Conference on Computer and Information Technology*, pp. 185–190, Sydney, NSW, Australia, 2008.
- [2] D. Aebi and L. Perrochon, *Towards Improving Data Quality*, In CiSMOD, 1993.
- [3] G. T. Reddy, M. P. K. Reddy, K. Lakshman et al., "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54776–54788, 2020.
- [4] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill Book Co, New York, 1983.
- [5] Y. Li and L. Bo, "A normalized Levenshtein distance metric," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1091–1095, 2007.
- [6] H. Galhardas and D. Florescu, "An extensible framework for data cleaning," in *Proceedings of the 16th IEEE International Conference on Data Engineering*, pp. 312–312, San Diego, California, 2000.
- [7] N. Deepa, Q. V. Pham, D. C. Nguyen et al., "A survey on blockchain for big data: approaches, opportunities, and future directions," 2020, <https://arxiv.org/abs/2009.00858>.
- [8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, Columbus, Ohio, the United States, 2014.
- [9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," 2015, <https://arxiv.org/abs/1506.01497>.
- [10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision Meets Robotics: The KITTI Dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [11] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: an evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2012.
- [12] S. Zhang, R. Benenson, and B. Schiele, "City persons: a diverse dataset for pedestrian detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Hawaii, the United States, 2017.
- [13] M. K. Singh, U. S. Tiwary, and Y. H. Kim, "An adaptively accelerated Lucy-Richardson method for image deblurring," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, 10 pages, 2007.
- [14] E. Nursultanov, M. Ruzhansky, and S. Tikhonov, "Nikolskii inequality and Besov, Triebel-Lizorkin, Wiener and Beurling spaces on compact homogeneous manifolds," 2014, <https://arxiv.org/abs/1403.3430>.
- [15] R. Girshick, "Fast r-cnn," in *In proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, Piscataway, NJ, 2015.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [17] A. R. Javed, M. Usman, S. U. Rehman, M. U. Khan, and M. S. Haghghi, "Anomaly detection in automated vehicles using multistage attention-based convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [18] A. Rehman, S. U. Rehman, M. Khan, M. Alazab, and T. Reddy, "CANintelliIDS: detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1456–1466, 2021.
- [19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, Las Vegas, Nevada, the United States, 2016.
- [20] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018, <https://arxiv.org/abs/1804.02767>.
- [21] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, Cham, 2016.
- [22] H. Law and J. Deng, "Cornersnet: detecting objects as paired keypoints," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 734–750, Munich, Germany, 2018.
- [23] X. Zhou, J. Zhuo, and P. Krahenbuhl, "Bottom-up object detection by grouping extreme and center points," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 850–859, California, the United States, 2019.