A New Method for Improving the Performance of YOLOv5 in Single Fuzzy Image Recognition

Yuhui Mao*

Department of Computer Science, SWJTU-Leeds Joint School, Southwest Jiaotong University, Chengdu, China

*Corresponding author: sc21ym@leeds.ac.uk

Abstract. Nowadays, multi-target detection technology has achieved high levels of accuracy and has been widely adopted in various domains. However, it is challenging for the multi-object detection model to process fuzzy images and return accurate results in adverse weather conditions such as rainy and foggy. In this paper, a preprocessing method designed specifically for images in rainy and foggy weather is introduced. The reduced visibility in these images arises from the scattering of light by water vapor, which blurs the boundary lines between objects and makes it challenging for the model to accurately identify the object's features. The function of the preprocessing model introduced in this paper is to enhance the distinction between objects in the image, thereby enhancing the object contour and improving the detection rate of the multi-object detection model. The preprocessing model in this paper is calculated based on the third-order matrix composed of image pixels and adjusts the brightness of pixels by calculating the brightness distribution of pixels between rows and columns, so as to realize the increase of the contrast between pixels within a certain threshold range, achieving the feature of strengthening the object boundary while protecting the integrity of the picture. The data and test in this experiment were all based on the model trained by YOLOv5, a representative model for multi-object recognition. The experimental results demonstrated that the preprocessing model introduced in this paper can enhance the recognition ability of YOLOv5 model and improve the recognition rate of YOLOv5.

Keywords: Image preprocessing; Multi-object detection; Image enhancement.

1. Introduction

Multi-object recognition is an application of deep learning network based on Convolutional Neural Network (CNN) computer vision research. The CNN process involves gradually abstracting the input image by filtering out excess image features to reduce the amount of redundant data while retaining the general features of the image [1, 2]. Up to now, due to the gradual strength of GPU computing power and the growing development of deep learning, multi-object recognition has been able to identify a variety of objects with high accuracy. Therefore, multi-object recognition technology is widely used in transportation, medical treatment and other fields. Among all the multi-object recognition models, YOLOv5 is the most representative, because YOLO series models have the nature of fast processing and high accuracy [3, 4], and are widely used in a variety of scenes. However, the accuracy of these multi-object recognition models in processing fuzzy images will greatly decrease, especially in the common rainy or foggy environment. Therefore, discussing how to deal with rainy and foggy days and other fuzzy images has become a topic worthy of discussion in computer vision. At present, many fuzzy concrete treatment methods for rainwater or haze are based on CNN and other networks to identify the location of rainwater and then remove it [5-7]. However, this kind of method will increase the depth of neural network to some extent and reduce the efficiency of computer processing model. In addition, the common point of these models is that most of them aim to remove raindrops so as to achieve clear images and enhance the ability of image processing. However, the method introduced in this paper is different from the idea of removing raindrops but focuses on strengthening the original outline of the image object, so as to help the model to better identify. This paper proposes an image preprocessing method for fuzzy images in rainy and foggy days and takes YOLOv5 model as an example to test the feasibility of this image preprocessing method.

In the process of recognizing special scenes such as rain and fog, the multi-object recognition model based on CNN is blocked by water vapor and refracted, resulting in a large number of light spots on the screen, leading to the blurring of the boundary between objects and scenes in the image. The fuzzy boundary will lead to the loss of some shape features after CNN layer abstracts image features. With the abstraction of each CNN layer, the deeper the network receives less information. Therefore, it is difficult for CNN to extract the feature information of objects, leading to recognition failure and declining accuracy. Therefore, how to restore the boundary blur caused by light spots is the focus of the image preprocessing method discussed in this paper. Since light spots will reduce the differentiation between pixels in the image, it is not difficult to find that the gradient of the object surface is relatively flat by converting the image into pixel gradient map under normal circumstances, while the pixel gradient between objects is relatively high. Therefore, the goal of the image preprocessing model proposed in this paper is to improve the distinction between pixels to achieve the function of highlighting the object boundary, so as to reduce the error caused by halo. The image preprocessing model introduced in this paper is different from other neural network algorithms based on the principle of rain or blurred image restoration. The model will be based on the basic 3x3 matrix calculation to highlight the line between light and dark in the image, so as to achieve the effect of segmentation between objects in the blurred image in rain and fog. This paper will introduce the image preprocessing method proposed in detail.

2. Methodology

2.1. Primary purpose

The image processing method proposed in this paper is to optimize the image with fuzzy objects in rain and fog. Many photos taken on rainy or foggy days have one feature in common: water mist, splash, or raindrop on rainy days can cause light to scatter, causing light from the background environment to enter the image of the object in the photo. The experiment was carried out based on the officially trained YOLOv5 model. After testing these images in rainy and foggy days, it is found that many objects that cannot be recognized in rainy day photos are mainly due to the interference of the boundary color of objects after the scattering of various ambient light, which leads to the gradient drop within a certain pixel range, resulting in the boundary color of objects being similar to the environment color. The purpose of the image preprocessing method introduced in this paper involves the extraction of a specific image region, followed by the assessment of pixel discrimination, which is then enhanced within the discrimination interval.

2.2. Proposed preprocessing method

By processing the image with a third-order matrix, the brightness of three lines of pixels is calculated respectively and the difference between them is also calculated. The increasement depends on the difference between rows and is multiplied by a weight. Users can adjust the brightness of pixels in the picture by adjusting the size of the weight. Taking the middle line as the benchmark, the pixel brightness is adjusted based on the increasement obtained. Because the distribution of pixels is complex, it is necessary to adjust the brightness of each pixel from two directions: vertical and horizontal.

2.3. Elaboration of the pre-process

The image preprocessing introduced in this paper consists of two parts, which are vertical preprocessing and horizontal preprocessing.

2.3.1. Vertical distribution direction preprocessing

First a 3-order matrix should be extracted from the picture (1).

$$\begin{matrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23}. \\ X_{31} & X_{32} & X_{33} \end{matrix} \quad (1)$$

The brightness of a pixel is denoted by $X_{ij}, \mathrm{i} \in \{1,2,3\}, \mathrm{j} \in \{1,2,3\}$.

To compute in one row of the matrix, the sum of each row needs to be computed (2)(3)(4):

$$L_{top} = X_{11} + X_{12} + X_{13}. \quad (2)$$

$$L_{cen} = X_{21} + X_{22} + X_{23}. \quad (3)$$

$$L_{bot} = X_{31} + X_{32} + X_{33}. \quad (4)$$

The sum of the top row L_{top}, the sum of the center row L_{cen}, and the sum of the bottom row L_{bot}, are calculated.

Then calculate the difference (5)(6):

$$\Delta L_{tc} = L_{top} - L_{cen} \quad (5)$$

$$\Delta L_{cb} = L_{cen} - L_{bot} \quad (6)$$

The purpose of this model is to calculate all the pixels in a third-order matrix and make the low brightness pixels less bright and the high brightness pixels brighter. In order to achieve this function, the increment of adjusting the pixel brightness should be calculated (The increment can be negative).

The goal of the model is to increase the discrimination between bright and dark pixels, therefore it is expected that pixels with a small difference will receive a larger increase in discrimination, while pixels with a large difference will receive a smaller or no increase.

A Sigmoid function shown in **Fig. 1** is a mathematical function which has a characteristic S-shaped curve [8]. $sigmoid - 0.5$ can meet the need of the pre-processing method, which has an intercept of 0. The slope of the sigmoid function is maximized when t is close to 0 and tends to 0 when $t = \pm 5$.

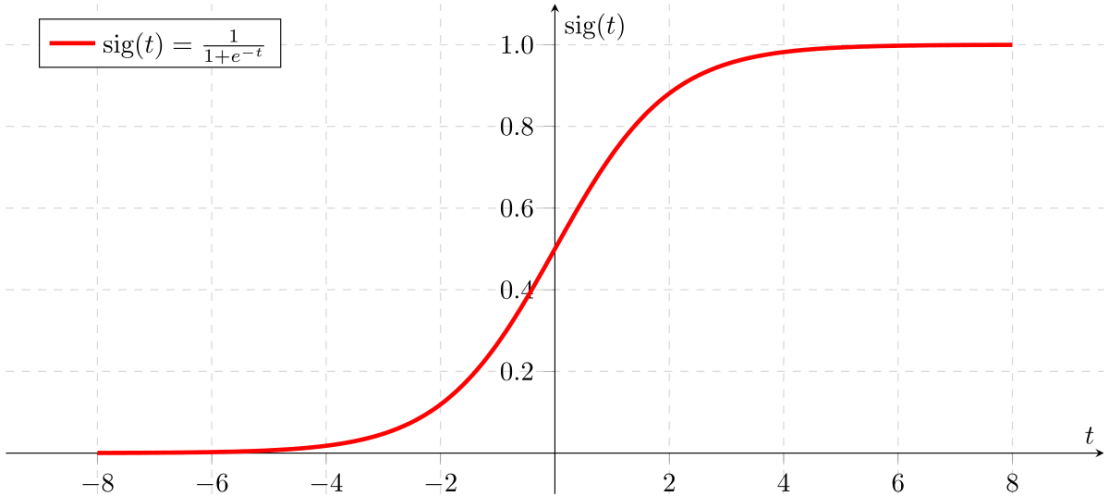

Fig. 1 Standard sigmoid function [9].

The increment is calculated as (7)(8):

$$Increasement_1 = [1 - \left(sigmoid\left(\frac{\Delta L_{tc}}{\theta}\right) - 0.5\right)] * weight \quad (7)$$

$$Increasement_2 = [1 - \left(sigmoid\left(\frac{\Delta L_{cb}}{\theta}\right) - 0.5\right)] * weight \quad (8)$$

Weight is a custom parameter that allows user to adjust the size of the increment. θ is an operational parameter that adjusts the sensitivity of the incremental computation equation to numerical values. Under normal conditions, $\theta = 25$, because the pixel brightness range is 0 to 255,

the difference is 255 or less, and the sigmoid function has an obvious slope from -5 to 5, with a span of 10, θ can be calculated as (9):

$$\theta = \frac{255}{10} = 25.5 \approx 25 \tag{9}$$

Finally, the Increment (INC) is added to the corresponding pixel row (10):

$$result_{vertical} = \begin{matrix} X_{11} + INC_1 & X_{12} + INC_1 & X_{13} + INC_1 \\ X_{21} & X_{22} & X_{23} \\ X_{31} + INC_2 & X_{32} + INC_2 & X_{33} + INC_2 \end{matrix} \tag{10}$$

2.3.2. Horizontal distribution direction preprocessing

The process in the horizontal direction is similar to that in the vertical direction, and the method of calculating rows is replaced by the method of calculating columns.
First, the sum of each column should be calculated (11)(12)(13):

$$L_{left} = X_{11} + X_{21} + X_{31} \tag{11}$$

$$L_{centre} = X_{12} + X_{22} + X_{32} \tag{12}$$

$$L_{right} = X_{13} + X_{23} + X_{33} \tag{13}$$

Next, the difference between columns ΔL_{lc} and ΔL_{cr} should be calculated as (14)(15):

$$\Delta L_{lc} = L_{left} - L_{centre} \tag{14}$$

$$\Delta L_{cr} = L_{centre} - L_{right} \tag{15}$$

Then the result of horizontal direction pre-process can be calculated by taking ΔL_{lc} and ΔL_{cr} into the function (16) (17)(18):

$$Increasement_1 = [1 - \left(sigmoid\left(\frac{\Delta L_{lc}}{\theta}\right) - 0.5\right)] * weight \tag{16}$$

$$Increasement_2 = [1 - \left(sigmoid\left(\frac{\Delta L_{cr}}{\theta}\right) - 0.5\right)] * weight \tag{17}$$

$$result_{horizontal} = \begin{matrix} X_{11} + INC_1 & X_{12} & X_{13} + INC_2 \\ X_{21} + INC_1 & X_{22} & X_{23} + INC_2 \\ X_{31} + INC_1 & X_{32} & X_{33} + INC_2 \end{matrix} \tag{18}$$

2.3.3. End of one 3×3 matrix pre-process

The result produced by the final model is $result = result_{horizontal} + result_{horizontal}$. This is the end of one 3×3 matrix pre-process.

In the implementation, it is necessary to determine a threshold value in the calculation of the ΔL_{lc} and ΔL_{cr} to prevent the incremental operation between pixels with the same brightness, as the pixels with the similar brightness are more likely to be the surface feature points of the same object, if the incremental operation will lead to the destruction of the image.

3. Result and discussion

The experiment involves the utilization of YOLOv5 models in an empirical investigation. Specifically, the objective of this experiment is to examine and evaluate the enhancement of a multi-object detection model through the implementation of a preprocessing technique proposed in this paper. The experiment entails a comparative analysis between the outcomes of YOLOv5 object detection on the original image and the pre-processed image.

3.1. Test example-1

Fig. 2 The detected result of YOLOv5 with the original image-1 (source from [10])

Fig. 2 is an image detected by the YOLOv5 model. It can be observed that the model does not detect the vehicle in the figure, or the recognition degree of the vehicle in the figure is not high enough, and the model does not make any reaction.

Fig. 3 The detected result of YOLOv5 with the pre-processed image-1

3.2. Following a comparative examination of Fig. 2 and Fig. 3, it can be discerned that the YOLOv5 model effectively recognizes the vehicle within the foggy scenery subsequent to image preprocessing. Nonetheless, the vehicle depicted on the left-hand side of the figure has experienced impairment due to the illumination from its own headlights, consequently compromising its integrity and rendering it challenging to be accurately restored and identified.Test example-2

Fig. 4 The detected result of YOLOv5 with the original image-2 (source from [11])

Fig. 5 The detected result of YOLOv5 with the pre-processed image-2

3.3. Fig. 5 illustrates a noteworthy enhancement in the identification of the majority of vehicles compared with Fig. 4. Nonetheless, it is important to acknowledge that the predicament whereby the roadside obstacles are erroneously categorized as individuals remains challenging to address, partly due to the constraints of the image and the inherent limitations of the model training process.Discussion

4. The present study highlights the utilization of a model to facilitate the completion of image preprocessing in the context of YOLOv5, resulting in a notable improvement in the accuracy of object detection. However, it is crucial to underscore that the testing process also revealed certain challenges associated with this approach. Specifically, while the image preprocessing techniques implemented were successful in augmenting the recognition capabilities of YOLOv5, it is noteworthy that there was a concomitant rise in the occurrence of false recognition. These findings underscore the importance of considering the trade-offs inherent in the adoption of such methods and the necessity of refining the approach to ensure optimal results.Conclusion

This paper introduces an image preprocessing method to enhance the object recognition ability of YOLOv5 in the context of fuzzy image processing, particularly in conditions such as rain and fog. By taking the third-order matrix as the unit to extract the image and then processing the pixels covered by the matrix through the mathematical method, the distinction between pixels on the image boundary can be effectively improved. In the experiment, objects blurred by rain were recognized by YOLOv5 algorithm under the action of the preprocessing method in this paper, and the object recognition rate in other pictures in rainy and fog weather was higher. However, it is essential to acknowledge that the experimentation also identified certain limitations of the proposed preprocessing algorithm. Specifically, although the method aided in improving the object recognition ability of YOLOv5, it simultaneously amplified the likelihood of error recognition, particularly in cases where the original image already had pre-existing recognition errors. Based on it, potential improvements in future research may be considered. For instance, by dividing the image into three RGB layers, appropriate weightage for each layer could be determined based on the physical scattering rate of the color in the water vapor. This could potentially enhance the effect of image preprocessing.

References

[1] Wikipedia Contributors, Convolutional neural network, Wikipedia, 2019. https://en.wikipedia.org/wiki/Convolutional_neural_network

[2] Chua, L.O., and T. Roska, The CNN paradigm, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 1993, 40(3): 147–156.

[3] Jiang, P., D. Ergu, F. Liu, Y. Cai, and B. Ma, A Review of Yolo Algorithm Developments, Procedia Computer Science, 2022, 199: 1066–1073.

[4] Shaifee, M.J., B. Chywl, F. Li, and A. Wong, Fast YOLO: A Fast You Only Look Once System for Real-time Embedded Object Detection in Video, Journal of Computational Vision and Imaging Systems, 2017, 3(1).

[5] Wan, Y., Y. Cheng, M. Shao, and J. Gonzàlez, Image rain removal and illumination enhancement done in one go, Knowledge-Based Systems, 2022, 252: 109244.

[6] Lin, K., S. Zhang, Y. Luo, and J. Ling, Unrolling Rain-guided Detail Recovery Network for Single Image Deraining, Virtual Reality & Intelligent Hardware, 2023, 5(1): 11–23.

[7] Xiao, J., M. Shen, J. Lei, J. Zhou, R. Klette, and H. Sui, Single image dehazing based on learning of haze layers, Neurocomputing, 2020, 389: 108–122.

[8] Wood, T., Sigmoid Function, DeepAI, 2019. https://deepai.org/machine-learning-glossary-and-terms/sigmoid-function

[9] Arunava, Derivative of the Sigmoid function, Medium, 2018. https://towardsdatascience.com/derivative-of-the-sigmoid-function-536880cf918e

[10] All About Vision, Driving in Fog: What Makes it Dangerous & How to Stay Safe, All About Vision. https://www.allaboutvision.com/eye-care/vision-road-safety/driving-in-fog/

[11] Lesschwab, Driving in Rain? How to Avoid Hydroplaning and Other Tips, www.lesschwab.com. https://www.lesschwab.com/article/driving-in-rain-how-to-avoid-hydroplaning-and-other-tips.html