Mathematical Biosciences and Engineering

MBE, 20(4): 7193–7216.
DOI: 10.3934/mbe.2023311
Received: 08 September 2022
Revised: 20 January 2023
Accepted: 31 January 2023
Published: 10 February 2023

http://www.aimspress.com/journal/MBE

Research article

Dynamic Mosaic algorithm for data augmentation

Yuhua Li[1], Rui Cheng[1], Chunyu Zhang[1], Ming Chen[1], Hui Liang[1,*] and Zicheng Wang[2,3]

[1] Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou 450001, China
[2] College of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450001, China
[3] Henan Key Lab of Information-Based Electronical Appliances, Zhengzhou University of Light Industry, Zhengzhou 450001, China

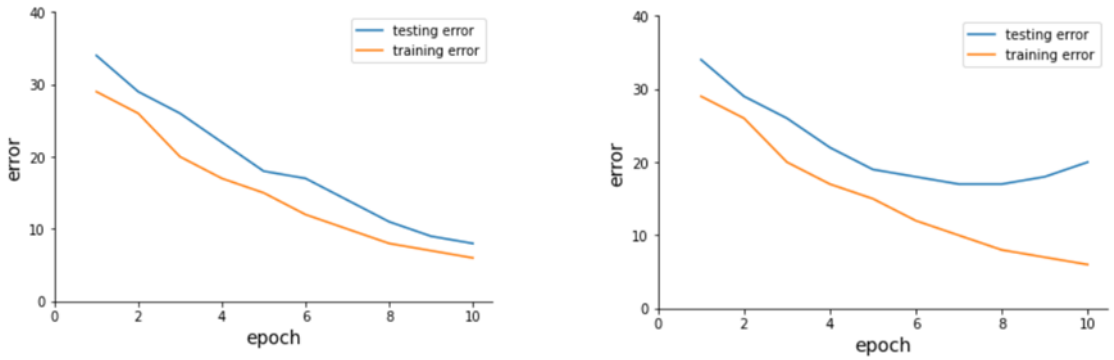
*** Correspondence:** Email: hliang@zzuli.edu.cn.

Abstract: Convolutional Neural Networks (CNNs) have achieved remarkable results in the computer vision field. However, the newly proposed network architecture has deeper network layers and more parameters, which is more prone to overfitting, resulting in reduced recognition accuracy of the CNNs. To improve the recognition accuracy of the model of image recognition used in CNNs and overcome the problem of overfitting, this paper proposes an improved data augmentation approach based on mosaic algorithm, named Dynamic Mosaic algorithm, to solve the problem of the information waste caused by the gray background in mosaic images. This algorithm improves the original mosaic algorithm by adding a dynamic adjustment step that reduces the proportion of gray background in the mosaic image by dynamically increasing the number of spliced images. Moreover, to relieve the problem of network overfitting, also a Multi-Type Data Augmentation (MTDA) strategy, based on the Dynamic Mosaic algorithm, is introduced. The strategy divides the training samples into four parts, and each part uses different data augmentation operations to improve the information variance between the training samples, thereby preventing the network from overfitting. To evaluate the effectiveness of the Dynamic Mosaic algorithm and the MTDA strategy, we conducted a series of experiments on the Pascal VOC dataset and compared it with other state-of-the-art algorithms. The experimental results show that the Dynamic Mosaic algorithm and MTDA strategy can effectively improve the recognition accuracy of the model, and the recognition accuracy is better than other advanced algorithms.

Keywords: data augmentation; mosaic algorithm; YOLOv5; deep learning; object detection

1. Introduction

Deep learning is a data-driven technique whose main purpose is to learn patterns and expressions in training samples, and is widely used in the field of computer vision [1–3], and trajectory outlier detection [4,5]. A recent study [6] has shown that the performance of deep learning models is logarithmically related to the number of training samples, which means that the larger the number of training samples, the better the generalization of the resulting model, and the better the performance. When the training samples are relatively small, overfitting is easy to occur in the actual application process [7,8]. Figure 1 depicts the visualization results of model error and epoch for ideal and overfitting cases. Among them, Figure 1(a) is the training process in an ideal situation. As the experiment proceeds, the error rates of both the train set and the test set are decreasing. Figure 1(b) shows the training process in the case of overfitting. The error rate of the test set first decreases and then increases with the iteration of the epoch. This is because the network memorizes the detailed features of the training samples, but these detailed features cannot be generalized [9,10]. Diversified training samples can prevent network overfitting [11,12], but the collection and production of training samples require a high cost, so low-cost and simple data augmentation [13,14] methods have become a more common choice for preventing network overfitting. In addition, data augmentation has the effect of reducing the sensitivity of the model to images and avoiding the unbalanced distribution of positive and negative samples [15], which is an effective method to improve the overall performance of the model.



(a) (b)

Figure 1. Comparison of the training process in the ideal and overfitting cases. (a) The ideal training process. (b) The training process of overfitting. The blue curve represents the error rate of the test set has a clear inflection point, which means that the model performs poorly on the test set relative to the training set.

Traditional data augmentation methods use random left-right flipping and cropping for training samples [16], which increases the diversity of training samples, and changes the brightness, saturation and contrast of images through color jitter. With the development of CNN, new network architectures have been proposed, such as AlexNet [17], VGG-16 [18], ResNet [19], DenseNet [20], etc. These architectures have deeper network layers, more complex structures and more parameters, so the risk of overfitting is also increasing [21]. The before-mentioned data augmentation techniques have been unable to effectively suppress the occurrence of overfitting.

Recently, researchers have done a lot of work in the field of mixed sample data augmentation [22],

among which the mosaic [23] algorithm proposed by Alexey et al. has achieved remarkable results. The mosaic algorithm mixes four training images and corresponding labels, so the generated mosaic images contain four different contexts, which not only allows the model to detect objects outside the normal context, but also helps to prevent overfitting of the network and improve the recognition accuracy of the model. In addition, since four images are stitched into one image, each layer can process the data of four images during the batch normalization calculates activation statistics [24] operation. This means that the mini-batch does not need to be very large to achieve good results, reducing the performance requirements for training equipment. However, the mosaic algorithm still has some shortcomings, the most typical shortcomings include: 1) There may be a large area of gray background not overwritten by the spliced images in the generated mosaic images, which will reduce the amount of information contained in the mosaic images. 2) Only mosaic images are used to participate in the training of the model, resulting in a single view of the training samples, and the information differences of the training samples are limited, which is not conducive to the generalization of the model.

Through in-depth research, we first aim to solve the problem of information waste caused by the large gray background in mosaic images. As such, this paper proposes a Dynamic Mosaic data augmentation algorithm. The Dynamic Mosaic algorithm adds a dynamic adjustment step based on the original mosaic [23] algorithm. By dynamically increasing the number of spliced images, it reduces the proportion of worthless areas in the mosaic image, and increases the complexity of the image content. Second, in order to solve the problem of the network overfitting and a single view of training samples, this paper also proposes a Multi-Type Data Augmentation (MTDA) strategy based on the Dynamic Mosaic algorithm. The MTDA strategy randomly divides the training samples into four groups, and each group of training samples is processed with different data augmentation techniques, thereby improving the information variance between the training samples.

In short, the main contributions of this paper include the following aspects:

• An improved mosaic data augmentation algorithm is proposed. The Dynamic Mosaic algorithm increases the dynamic adjustment step on the basis of the original mosaic algorithm, reduces the proportion of the worthless area in the mosaic image, and improves the quality of the generated mosaic image.

• A data augmentation strategy is introduced. The MTDA strategy divides the training sample into multiple parts, and each part uses different data augmentation operations to improve the information difference between the input images and solve the problem of overfitting caused by a single view of the training sample.

The rest of this paper is organized as follows: In the next section, we give a brief review of related work about data augmentation methods. In Section 3, we describe the method proposed in this paper in detail. In Section 4, we report the experimental results of the Dynamic Mosaic algorithm and the Multi-Type Data Augmentation strategy on the Pascal VOC dataset. Finally, we conclude the paper in Section 5.

2. Related works

Data augmentation is an effective regularization [25] method, which can increase the number and diversity of training samples to improve the generalization ability of the model and prevent network overfitting. This paper divides related work into four categories: traditional data augmentation methods, data disrupting methods, unsupervised data augmentation methods and mixed sample data augmentation methods.

In traditional data augmentation methods, all operations focus on the samples themselves, mainly based on the data morphology of the image for data augmentation, including flip, rotation, crop, zoom and color change, and other operations [26,27]. These techniques have been shown to be useful for specific datasets, for example, random cropping and horizontal flipping techniques are very helpful for the recognition task of the CIFAR dataset [28]. However, only using these transformation methods will result in a single data sample, and cannot effectively suppress the occurrence of overfitting [29]. Therefore, researchers have proposed more advanced data augmentation methods from other perspectives.

Data disrupting is also a common data augmentation method, which randomly zeros out a part of the image to achieve the purpose of changing the characteristics of the original image. For example, the cutout [30] algorithm randomly crops a square patch in the image and replaces it with "0" pixels, as shown in Figure 2. Noise is introduced into the image by masking, which makes the CNN more robust to noisy images. Furthermore, when the patch masks the main part of the object, such as the cat head, in this case, the CNN needs to learn the rest of the cat (such as ears and paws) to recognize the object. This method improves the utilization of minor features in the images, which is helpful for improving the recognition accuracy of the model. However, the data disrupting method will cause the loss of pixel information of the images.

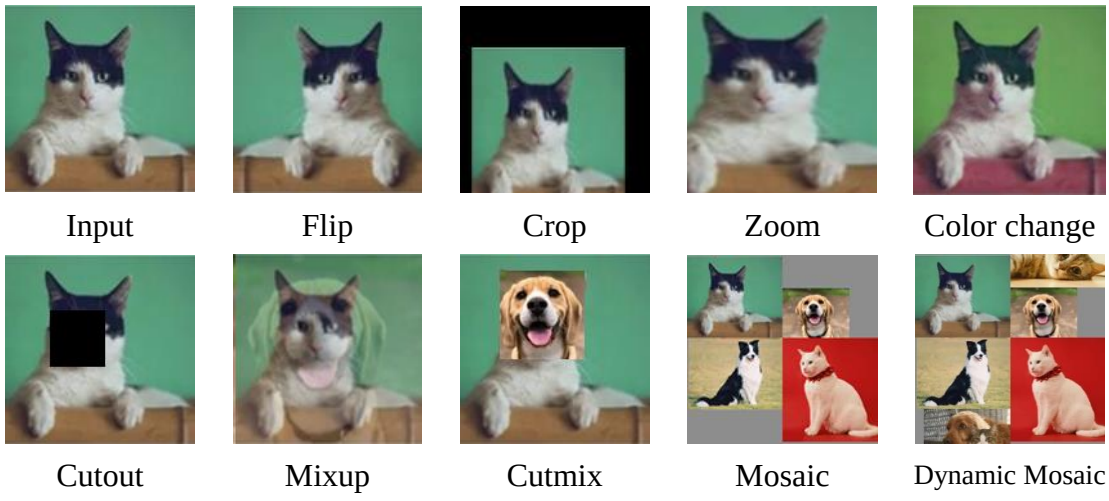


Figure 2. Visual comparison of different data augmentation techniques.

Unsupervised data augmentation methods are mainly divided into two categories: autoaugment [31] method and GAN [32] method. Autoaugment method generates a data augmentation strategy suitable for a specific dataset, but using this method to explore a data augmentation strategy takes a lot of time [33,34]. The GAN performs data augmentation operations by randomly generating images that are consistent with the distribution of training samples [35],

which increases training time by an order of magnitude and performs poorly on non-adversarial images in accuracy [36].

The idea of mixed sample data augmentation is to use multiple samples to generate new samples. For example, the mixup [37] algorithm improves the generalization ability of the model by performing convex linear interpolation [38] on two images according to a certain ratio, and then fusing them into a new training sample. As seen in Figure 2, however, mixup images are blurry and unnatural in the representation of some local features [39]. The cutmix [40] algorithm superimposes the cropped region of another input image onto the patch region, which solves the problem of loss of pixel information in the cutout algorithm. The mosaic algorithm has a certain similarity with the cutmix algorithm in theory. The cutmix algorithm is to crop and stitch the 2 images in the dataset, and the mosaic algorithm is to stitch the four images in the dataset into a new image.

3. Methods

In this section, we describe, in detail, the Dynamic Mosaic data augmentation algorithm and MTDA strategy proposed in this paper. First of all, the Dynamic Mosaic algorithm adds a dynamic adjustment step on the basis of the original mosaic algorithm. By dynamically adjusting the number of spliced images, the proportion of worthless areas in the mosaic image is reduced and the complexity of the image content is increased. In addition, the MTDA strategy uses the mixup algorithm and the cutout algorithm on the basis of the Dynamic Mosaic algorithm. Further, the training samples are divided into four parts, where each part uses different data augmentation operations to increase the information variance between the training samples, thereby preventing the network from overfitting.

3.1. Dynamic Mosaic algorithm

There are two main purposes of using multiple image stitching: one is to increase the complexity of the image content, and the other is to increase the number of target objects in the image [41]. Both ways can motivate the trained model to have better detection performance and generalization ability. The idea of the mosaic algorithm is to randomly select four images, takes parts of them and stitches them into a mosaic image, and the excess part will be discarded. The mosaic algorithm mainly includes five image processing steps: First, randomly select the indices of four images and form the $K \in \{1,2,3,4\}$. Second, initialize the mosaic image. It should be noted that the size of the mosaic image is twice the size of the input image. For example, the shape of the input image is $640 \times 640 \times 3$, then the shape of the created mosaic image should be $1280 \times 1280 \times 3$. Third, use a random function to obtain a center point splicing coordinate (x_c, y_c) on the created mosaic image. Fourth, place the spliced images into the mosaic image in the order of upper left, upper right, lower left and lower right around the center point splicing. Finally, the periphery of the mosaic image is cropped to obtain a mosaic image with a shape of $640 \times 640 \times 3$. An example of the mosaic algorithm is shown in Figure 3.

The original mosaic image is always composed of four spliced images, but it is affected by the position of the center point splicing and the size of the spliced image itself. When the position of the center point splicing is close to the border of the mosaic image or the size of the spliced image itself is small, the spliced image cannot overwrite the entire area, and a large area of gray background is easy to appear in the generated mosaic image, as shown in Figure 4. The gray background is created

when the mosaic image is initialized and appears in the mosaic image because it is not overwritten by the spliced image. There are no valuable target objects in the gray background, so when a large area of gray background appears in the mosaic image, it reduces the amount of information contained in the mosaic image. To solve the above problems, the Dynamic Mosaic algorithm is proposed. By dynamically increasing the number of spliced images and overwriting the "worthless" gray background with new images, the problem of information waste caused by the appearance of gray background can be solved. Moreover, the Dynamic Mosaic algorithm can increase the complexity of the image content and increase the number of target objects in the image to a certain extent.

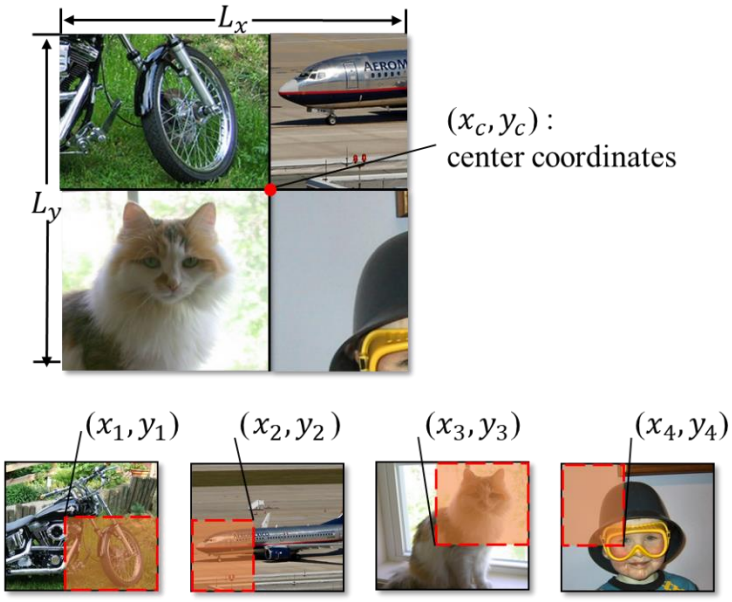


Figure 3. An example of the mosaic algorithm. By splicing four images, not only is the diversity of training samples increased, but the number of target objects is also increased.

In the Dynamic Mosaic algorithm, the number of spliced images is not a fixed value, but is determined by judging the distance from the border of spliced images to the border of the mosaic image after each stitching operation is completed; among them, this distance is represented by pad_h and pad_w. When the distance from the border of spliced image to the border of the mosaic image exceeds the threshold τ, the Dynamic Mosaic algorithm will acquire another image from the dataset and overwrite it on a gray background, which we call the re-acquired image the respliced image. The coordinates of the upper left and lower right corners of the respliced image in the mosaic images are represented by (a_i,b_i) and (c_i,d_i), and the positions of the coordinates (a_i,b_i) and (c_i,d_i) are affected by pad_h and pad_w. When $\text{pad}_h \geq \tau$ and $\text{pad}_w \leq \tau$, the calculation process of the upper left corner coordinate (a_i,b_i) of the respliced image is shown in Eq (1).

$$a_i = \begin{cases} \max(x_c - w_2, 0), & i=1,3 \\ x_c, & i=2,4 \end{cases}$$

$$b_i = \begin{cases} \max(y_c + h_1 + h_2, 0), & i=1,2 \\ y_c - h_1, & i=3,4 \end{cases} \tag{1}$$

The calculation process of the lower right corner coordinate (c_i,d_i) of the respliced image is shown in Eq (2).

$$c_i=\begin{cases} x_c, & i=1,3 \\ \min(x_c+w_2\ ,\ s)\ , & i=2,4 \end{cases}$$

$$d_i=\begin{cases} y_c+h_1, & i=1,2 \\ \max\left(y_c-h_1-h_2\ ,-s\right)\ , & i=3,4 \end{cases} \tag{2}$$

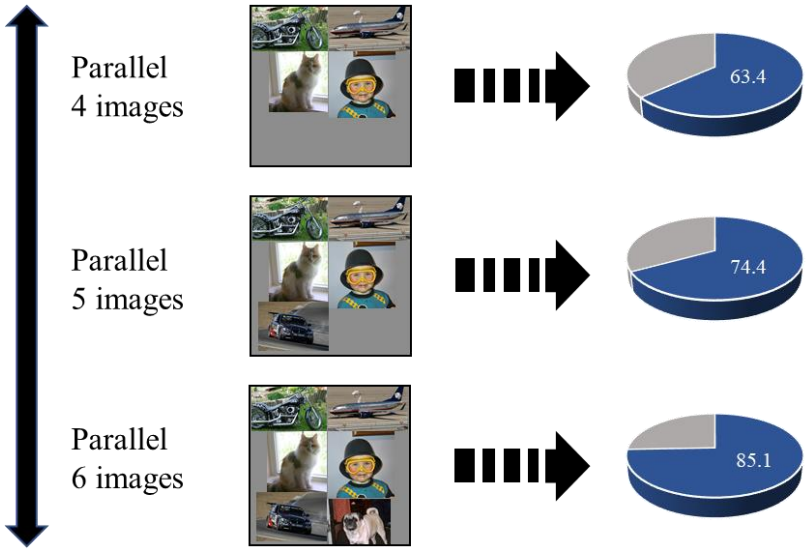


Figure 4. An example of the Dynamic Mosaic algorithm. The blue part represents the proportion of the spliced image, and the gray part represents the proportion of the "worthless" gray background. By increasing the number of images involved in spliced, the problem of information waste caused by the appearance of large-area gray backgrounds can be solved.

where, x_c and y_c are the abscissa and ordinate of the center point splicing respectively, h_1 and w_1 are the length and width of the spliced image, h_2 and w_2 are the length and width of the respliced image, s is the size of the mosaic image, i from 1 to 4 represents the upper left, upper right, lower left and lower right of the mosaic image, respectively, max(a, b) means taking the maximum value among a and b, and min(a, b) means taking the minimum value among a and b.

When $\text{pad}_h \le \tau$ and $\text{pad}_w \ge \tau$, the calculation process of the upper left corner coordinate (a_i,b_i) of the respliced image is shown in Eq (3).

$$a_i=\begin{cases} \max(x_c-w_1-w_2\ ,\ 0)\ , & i=1,3 \\ x_c+w_1, & i=2,4 \end{cases}$$

$$b_i=\begin{cases} \max\left(y_c+h_2\ ,\ 0\right)\ , & i=1,2 \\ y_c, & i=3,4 \end{cases} \tag{3}$$

The calculation process of the lower right corner coordinate (c_i,d_i) of the respliced image is shown in Eq (4).

$$c_i=\begin{cases} x_c\text{-}w_1, & i=1,3 \\ \min(x_c+w_1+w_2\ ,\ s)\ , & i=2,4 \end{cases}$$

$$d_i=\begin{cases} y_c, & i=1,2 \\ \max\left(y_c\text{-}h_2\ ,\text{-}s\right), & i=3,4 \end{cases} \tag{4}$$

When $\text{pad}_h \geq \tau$ and $\text{pad}_w \geq \tau$, we will compare the magnitude of the pad_h and pad_w values and perform the dynamic adjustment step only in the larger dimension. When $\text{pad}_h < \tau$ and $\text{pad}_w < \tau$, we will not perform the dynamic adjustment step. In the Dynamic Mosaic algorithm, we do not completely remove the gray background in the mosaic image, because, in the prediction stage, the input image will be filled with a grey background to fit the dimensions of the input layer. Therefore, retaining a small amount of gray background in the training samples will help the model learn to ignore the gray background. A specific description of the Dynamic Mosaic algorithm is given in the pseudocode below.

Pseudocode 1: Dynamic Mosaic data augmentation algorithm

Input: Image I_{orig}, Labels L_{orig}, hyperparameter τ
function AugmentMosaic (Image I_{orig}, Labels L_{orig})
  x_c, y_c=random.uniform (β_1, β_2)
  Creating mosaic images I_{dm}.size (2*s, 2*s).color (114)
  for i = 1, 2, ⋯, k **do**
    $I_{dm}[(xa1_i, ya1_i), (xa2_i, ya2_i)] \leftarrow I_{orig}[(xb1_i, yb1_i), (xb2_i, yb2_i)]$
    $\text{pad}_w=|xa1_i\text{-}xb1_i|$, $\text{pad}_h=|ya1_i\text{-}yb1_i|$
    if($\text{pad}_w \geq \tau$ or $\text{pad}_h \geq \tau$)
      $I_{dm}[(rxa1_i, rya1_i), (rxa2_i, rya2_i)] \leftarrow I_{orig}[(rxb1_i, ryb1_i), (rxb2_i, ryb2_i)]$
    Labels $L_{dm} \leftarrow$ Labels L_{orig}
  end for
  clip Images I_{dm}, Labels L_{dm}
end function
Output Images I_{dm}, Labels L_{dm}

3.2. Multi-Type data augmentation strategy

Although the information difference of each image is improved by means of the Dynamic Mosaic algorithm, the information variance between images is not greatly improved. This may cause the model to have higher recognition accuracy when recognizing images with similar distributions of testing samples and training samples, but when the distributions of testing samples and training samples do not match, the recognition accuracy of the model will not meet the needs of the task.

In order to improve the information variance between training samples, this paper proposes the MTDA strategy. The MTDA strategy randomly divides the input image into four parts, and uses different data augmentation methods for each part of the image. Finally, the obtained training samples include original image, mosaic image, mosaic-mixup image and mosaic-cutout image. Among them, the mosaic image is generated after using the Dynamic Mosaic algorithm, the mosaic-mixup image is generated after the Dynamic Mosaic and mixup algorithms, and the

mosaic-cutout image is generated after using the Dynamic Mosaic and cutout algorithms. An example of the MTDA strategy is shown in Figure 5. For the convenience of representation, the four types of images are represented by D_{og}, D_{dm}, D_{mm} and D_{mc}, respectively, where the relationship between the augmented output image D_{out} and the four types of images can be represented as $D_{out}=D_{og}UD_{dm}UD_{mm}UD_{mc}$, and the ratios of the number of samples in them are $W_1:W_2:W_3:W_4$.

The D_{mm} image is superimposed with the mixup algorithm based on the Dynamic Mosaic algorithm. The mixup algorithm performs convex linear interpolation on the two images according to a certain ratio, and then fuses them into a new sample. An example of the mixup algorithm is shown in Figure 6, and the calculation process is shown in Eq (5).

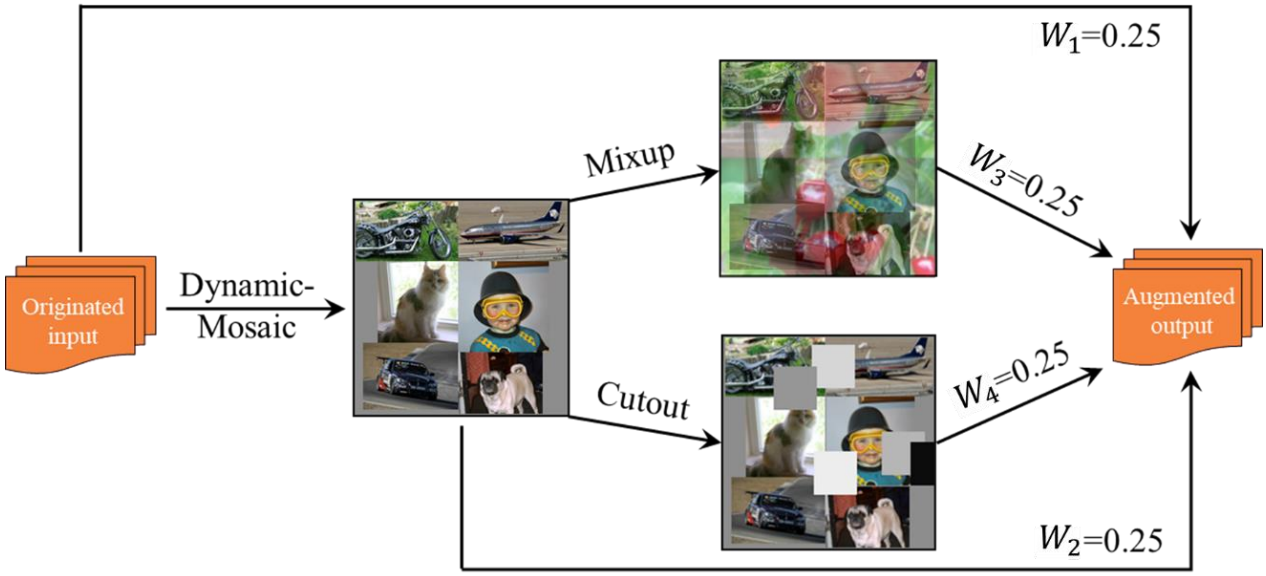


Figure 5. An example of Multi-Type Data Augmentation strategy. The MTDA strategy uses different types of training samples, which increases the information variance between training samples, so it can better prevent network overfitting.

$$x=\lambda x_i+(1-\lambda)x_j$$
$$y=\lambda y_i+(1-\lambda)y_j \tag{5}$$
$$\lambda\in\text{Beta}(\beta,\beta)$$

where x_i and x_j represent the two images involved in the fusion, y_i and y_j represent the labels of the two images, x and y represent the mixed images and labels after fusing the x_i and x_j images. λ and β are real numbers, and $\lambda\in[0, 1]$, $\beta\in(0,\infty)$, λ conforms to the Beta distribution. The D_{mm} image contains, not only the spatial blending feature after using the Dynamic Mosaic algorithm, but also the pixel blending feature after using the mixup algorithm, which is very helpful for improving the recognition accuracy of objects. The mixup algorithm, however, sometimes generates local features that do not exist in the original image, resulting in excessive adversarial interference. Therefore, the MTDA strategy uses four different types of images as training samples to participate in the training of the model, and the D_{mm} image is only used as part of the input image. This method not only solves the problem of excessive adversarial interference, but also the diverse training samples can prevent the network from overfitting, making the model perform better in challenging recognition and detection tasks.

The D_{mc} image first uses the Dynamic Mosaic data augmentation algorithm, and then uses the cutout algorithm on this basis. The idea of the cutout algorithm is to randomly select a center point in the image, and then use a patch to mask the image around the center point. This operation is a continuous dropout of input pixels to prevent the overfitting, and encourages the network to utilize information from the entire image rather than relying on a small subset of specific visual features. By superimposing the Dynamic Mosaic and cutout algorithms, it not only increases the complexity of the training sample content, but also enables the network to better combine the context around the noise and focus on some local secondary features. A specific description of the MTDA strategy is given in the pseudocode below.

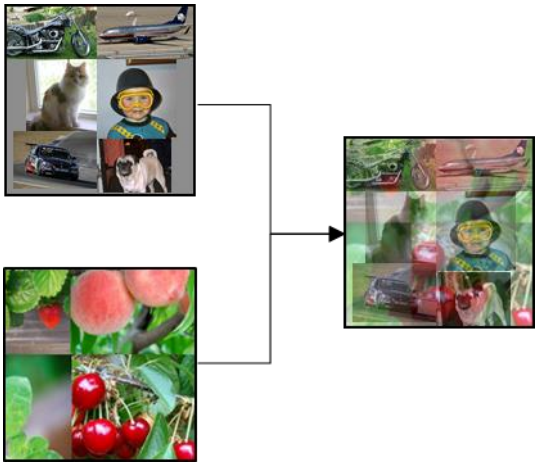


Figure 6. An example of the mixup algorithm. The two images are fused together by convex linear interpolation, which improves the linear representation between training samples.

Pseudocode 2: Multi-Type Data Augmentation strategy

1: **Input:** Image I_{orig}, Labels L_{orig}, hyperparameter W_2, W_3, W_4
2: **function** MTDAstrategy (Image I_{orig}, Labels L_{orig})
3: bool mosaic= random.random()<W_2
4: **if** (mosaic)
5: Images I_{mtda}, Labels L_{mtda} = loadmosaic (self, index)
6: **if** (random.random()<W_3)
7: Images I_{mtda}, Labels L_{mtda}=mixup(*loadmosaic (self, random.randint(0, self.n-1)))
8: **elif (**random.random()<W_4**)**
9: Images I_{mtda}, Labels L_{mtda}= cutout (image, labels)
10: **else** Images I_{mtda}, Labels L_{mtda}=loadimage (self, index)
11: **end function**
12: **Output** Images I_{mtda}, Labels L_{mtda}

In addition, we also verify the performance of the models combined with different data augmentation methods. Among them, the strategy combining the four data augmentation methods achieved the best performance. For the specific experimental results, please refer to Table 6 in Chapter 4.4.2.

4. Experiment

In this section, we first introduce the datasets used in the experiments and the evaluation metrics we adopted. Then, we describe the experimental environment of this paper and the parameter configuration in the experiment. In addition, we present the experimental results, analyze the data and give our conclusions. Finally, we discuss some interesting findings derived from the experimental data.

4.1. Datasets

In this paper, we conducted experiments and evaluate the Dynamic Mosaic algorithm and MTDA strategy on the Pascal VOC [42] dataset. The Pascal VOC dataset contains a total of 20 classes, which can be divided into four main categories: vehicles, household, animals and other. The specific classes are shown in Table 1. The Pascal VOC dataset contains two subsets: Pascal VOC 2007 and Pascal VOC 2012. Since the test set of Pascal VOC 2012 is not public, this paper uses the train set of Pascal VOC 2007 and Pascal VOC 2012 as the train set of the experiments, which contains 16551 images. The test set of Pascal VOC 2007 is used as the test set of our experiments, which contains 4952 images.

Table 1. The Pascal VOC classes.

Vehicles	Household	Animals	Other
Aeroplane	Bottle	Bird	Person
Bicycle	Chair	Cat	
Boat	Dining table	Cow	
Bus	Potted plant	Dog	
Car	Sofa	Horse	
Motorbike	TV/Monitor	Sheep	
Train			

4.2. Evaluation metrics

To evaluate the performance of the proposed algorithms, this paper uses precision, recall, F_1 score and mean Average Precision (mAP) as evaluation metrics to measure the recognition accuracy of the detection model. Among them, precision represents the correct proportion of the results predicted by the model, recall represents the proportion of the real target that the model predicts correctly. The calculation processes of precision and recall are shown in Eqs (6) and (7).

$$\text{precision}=\frac{\text{TP}}{\text{TP+FP}} \tag{6}$$

$$\text{recall}=\frac{\text{TP}}{\text{TP+FN}}, \tag{7}$$

where TP represents True Positive, which means the positive examples are correctly classified, FP represents False Positive, indicating that negative examples are incorrectly classified as positive

examples, and FN represents False Negative, which means that positive examples are incorrectly classified as negative examples. F_1 score takes into account the precision and recall of the model, and can be regarded as a harmonic average of the precision and recall of the model. The calculation process of F_1 score is shown in Eq (8).

$$F_1=2*\frac{\text{precision*recall}}{\text{precision+recall}} \tag{8}$$

mAP is the average of all classes of AP, the larger the mAP value, the higher the recognition accuracy of the model. The calculation process of mAP is shown in Eq (9).

$$\text{mAP}=\frac{1}{N}\sum_{i=1}^{N}\text{AP}_i, \tag{9}$$

where N is a constant representing the number of classes in the dataset, and AP_i represents the average accuracy for class i. In addition, unless otherwise specified, the default AP value is obtained when IoU=0.5.

4.3. Experiment environment and parameter configuration

All the experiments were performed on a desktop computer with the CPU Intel(R) Xeon(R) Gold5117, memory 128G, GPU NVIDIA Tesla V100, 16G video memory. The software configuration of the experimental platform is as follows: the operating system is Ubuntu18.04 (64-bit), the CUDA version is 10.2, the Python version is 3.8.3, and the open source neural network framework of Pytorch 1.10.0.

This paper uses YOLOv5s [43] as the detection model. In order to ensure the fairness of the experimental results, the model is trained from scratch for each experiment. The input image size is 640 × 640 × 3, the batch_size is 32, the initial learning rate is 0.01, the threshold τ for the dynamic adjustment method is 480, the β in mixup data augmentation is 8 and $W_1{:}W_2{:}W_3{:}W_4{=}1:1:1:1$. All experiments used a warm-up strategy for the first 3 epochs and a cosine annealing strategy [44] from the 4$^{\text{th}}$ epoch.

4.4. Experimental results and analysis

In this section, we first present and analyze the experimental results of the Dynamic Mosaic algorithm, which mainly includes the comparison of the experimental results of the Dynamic Mosaic algorithm and the mosaic algorithm, and the comparison of the experimental results of the Dynamic Mosaic algorithm and other state-of-the-art algorithms. Then, we show and analyze the experimental results of the MTDA strategy, which mainly include the comparison of the experimental results before and after the use of the MTDA strategy, the comparison of the experimental results of the MTDA strategy and different data augmentation strategy, and the comparison of the experimental results of different training sample combinations in the MTDA strategy.

4.4.1. Experimental results and analysis of Dynamic Mosaic

Increasing the complexity of the training sample background and the number of objects in the training sample is a reliable way to improve the detection performance and generalization ability of

the model. The mosaic image generated by the original mosaic algorithm only contains the pixel information of four spliced images and is affected by the position of the center point splicing and the size of the spliced image itself, and the area not overwritten by the spliced image will be occupied by "worthless" gray areas. According to the Dynamic Mosaic algorithm proposed in Section 3, by dynamically increasing the number of spliced images based on the original mosaic algorithm, the proportion of worthless regions in the mosaic image can be reduced, thereby improving the recognition accuracy of the model.

To demonstrate the effectiveness of the Dynamic Mosaic data augmentation algorithm, we conducted experiments on the Pascal VOC dataset. It should be noted that we selected the maximum value of mAP in 300 epochs, and recorded the precision and recall at the same time, and obtained the F_1 score by calculating the harmonic average of precision and recall. The experimental results are shown in Table 2.

Table 2. Experimental results of the Dynamic Mosaic algorithm and the mosaic algorithm.

Scheme	Method	P (%)	R (%)	F_1 (%)	mAP (%)
A	Original	76.03	65.05	70.11	71.19
B	Mosaic	78.99	**73.15**	75.96	79.01
C	Dynamic Mosaic (Our)	**80.13**	72.92	**76.36**	**79.73**

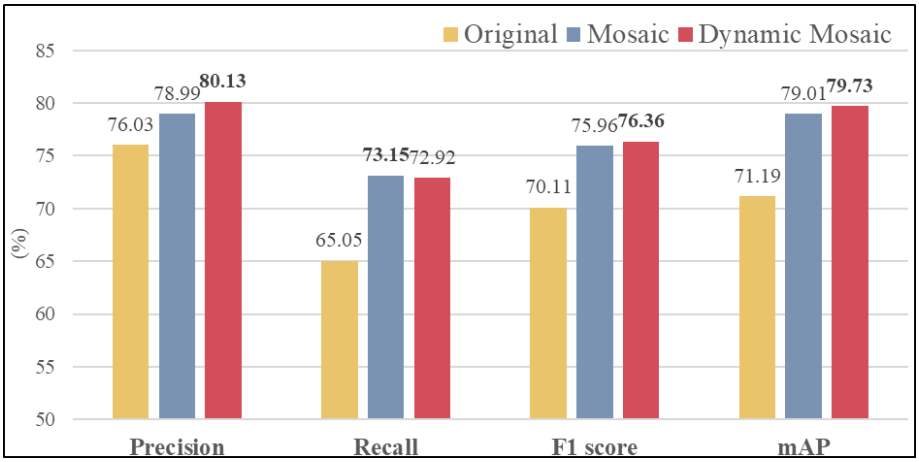


Figure 7. Comparison of experimental results of different schemes. Although the evaluation metrics recall of the Mosaic Scheme is slightly higher than that of the Dynamic Mosaic Scheme, compared with the Original and Mosaic Schemes, the Dynamic Mosaic Scheme has different degrees of improvement in the evaluation metrics precision, F_1 score and mAP. This is because the Dynamic Mosaic data augmentation algorithm adds a dynamic adjustment step based on the mosaic algorithm, which solves the problem of information waste in the original mosaic algorithm, and increases the complexity of the image content, so that the detection model has better generalization ability.

Among them, Scheme A does not use the data augmentation algorithm, and only uses the original image as the training sample to participate in the training of the model. Compared to Scheme B and Scheme C using data augmentation algorithms, Scheme A performs poorly. This shows that using a suitable data augmentation algorithm can improve the recognition accuracy of the model.

Scheme B uses the mosaic data augmentation algorithm. Compared with Scheme A, the evaluation metrics of Scheme B have been greatly improved, and the highest recall has been achieved among the three schemes. Scheme C uses the Dynamic Mosaic data augmentation algorithm proposed in this paper. It can be seen from the experimental results that, although the recall of Scheme C is lower than that of Scheme B, Scheme C has different degrees of improvement in the evaluation metrics precision, F_1 score and mAP. Among them, the evaluation metrics precision is improved by 1.14%, which indicates that using the Dynamic Mosaic data augmentation algorithm during the training process can improve the generalization ability of the detection model, thereby reducing the detection error rate of the model for some difficult samples. In addition, the F_1 score and mAP are improved by 0.4 and 0.72%, respectively, compared with the Scheme B, which indicates that the detection model using the Dynamic Mosaic data augmentation algorithm has better comprehensive performance. In order to more intuitively see the difference of the performance of each scheme, this paper draws the experimental results of each scheme, as shown in Figure 7.

In order to demonstrate the advancement of the Dynamic Mosaic data augmentation algorithm, this paper selects the more advanced algorithms for comparison, including cutmix [40], mixup [37] and mosaic9 [45] algorithms. The cutmix and mixup algorithms have been introduced in Chapter 2, so here we introduce the mosaic9 algorithm. The mosaic algorithm splices four images to generate a new image, and the mosaic9 algorithm splices nine images into a new image. The mosaic9 algorithm improves the recognition ability of the model in complex backgrounds and improves the accuracy of small objects recognition. The experimental results of all schemes are shown in Table 3. It can be seen that the training time of Scheme A using the original image is the shortest, which is 14.2 hours, and the training time of Scheme D is the longest, which is 25.6 hours. Scheme F, using the Dynamic Mosaic algorithm, achieves the highest recognition accuracy of 79.73%, and Scheme B has the lowest recognition accuracy of 69.62%. Compared with the mosaic algorithm before the improvement, although the training time of the Dynamic Mosaic data augmentation algorithm is increased by 0.7 hours, the recognition accuracy is improved by 0.72%. The experimental results show that compared with other advanced algorithms, the Dynamic Mosaic algorithm, proposed in this paper, has advantages in recognition accuracy.

Table 3. Comparison of recognition accuracy and training time of different algorithms.

Scheme	Method	mAP (%)	Training time (h)
A	Original	71.19	**14.2**
B	CutMix	69.62	15.4
C	Mixup	77.33	17.8
D	Mosaic9	78.97	25.6
E	Mosaic	79.01	15.7
F	Dynamic Mosaic (Our)	**79.73**	16.4

Figure 8 shows the detection examples of the mosaic algorithm and the Dynamic Mosaic algorithm on the Pascal VOC dataset. The first line is to use the original images as training samples without data augmentation. It can be seen from the detection examples that the confidence of the Original Scheme for the target prediction is relatively poor. For example, the confidence of the people in the second group of images is only 0.15. In addition, there are cases of missed detection in

the Original Scheme, such as the cat in the first group of images are not recognized. The second line is the scheme using the mosaic algorithm. It can be seen from the detection examples that the Mosaic Scheme has a false detection, for example, as the yellow dog in the third group of images is falsely identified as a sheep and has a confidence of 0.49. Also, the dog's leg is falsely identified as a cow and has a confidence of 0.46. The third line is the scheme using the Dynamic Mosaic algorithm proposed in this paper. The Dynamic Mosaic Scheme has no missed detection or false detection, and has higher confidence than the Original Scheme and the Mosaic Scheme, for example, as the confidence of cats in the first group of images increased from 0.37 to 0.56, and the confidence of horses in the second group of images increased from 0.37 to 0.70.

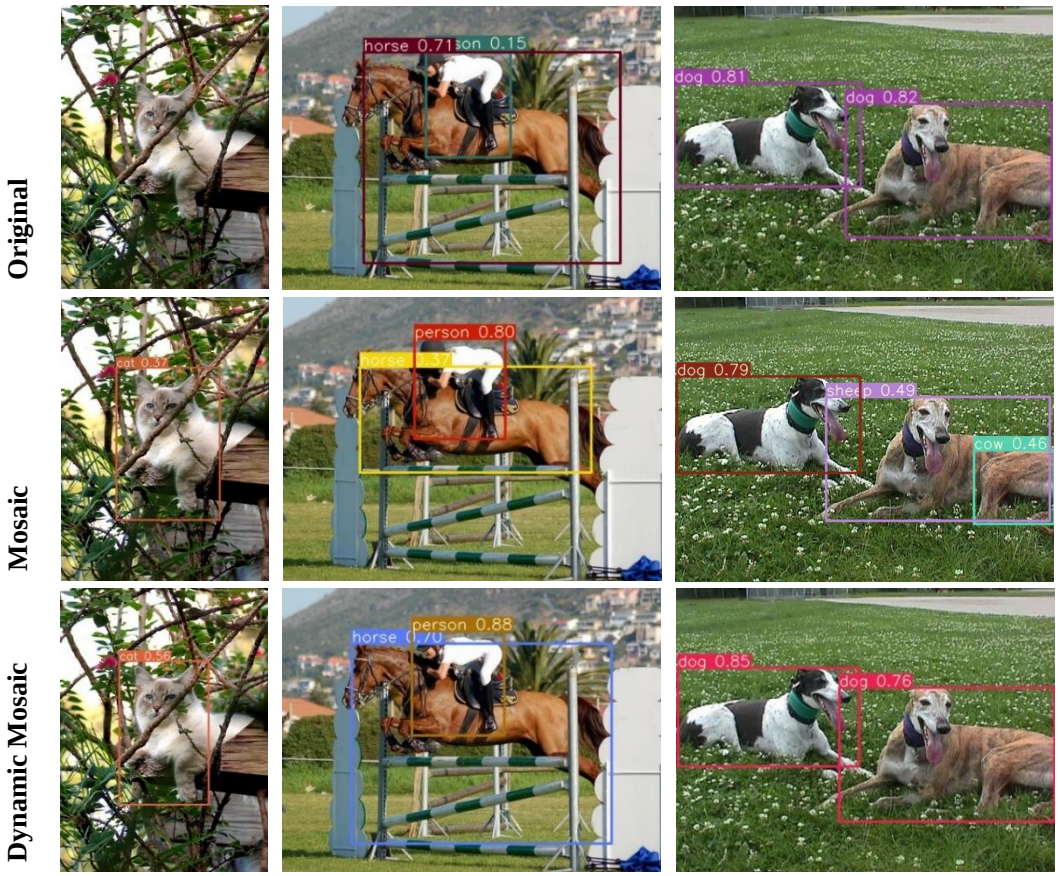


Figure 8. Comparison of detection examples of different data augmentation schemes. The detection example comes from the test set of Pascal VOC 2007. It can be seen from the detection results that the detection model using the Dynamic Mosaic algorithm can recognize the target object more accurately, and there is no missed detection or false detection.

4.4.2. Experimental results and analysis of MTDA strategy

Only using mosaic images that have been generated by the Dynamic Mosaic algorithm to participate in the training of the model will cause the training samples to maintain a single view, which limits the difference in information, and is not conducive to the generalization of the model. According to the MTDA strategy proposed in Section 3, by randomly dividing the training samples

into four parts, each part of training samples is processed with different data augmentation operations, which will increase the information variance between the training samples, and is beneficial to the generalization of the model and improves the recognition accuracy of the model.

To demonstrate the effectiveness of the MTDA strategy, we conducted experiments on the Pascal VOC dataset. The experimental results are shown in Table 4. Scheme A only uses the Dynamic Mosaic algorithm, and Scheme B additionally uses the MTDA strategy based on Scheme A. It can be seen, from the experimental results, that each evaluation metrics of Scheme B have been greatly improved. Among them, precision increased by 1.71%, recall increased by 3.53%, F_1 score increased by 2.69% and mAP increased by 3.68%. This is because the MTDA strategy divides the input image into four parts, and the samples of each part are processed with different data augmentation methods, which improves the information variance between training samples. In addition, the diverse training samples can improve the generalization and robustness of the model and prevent the network overfitting, so that the model can have better recognition effect on complex and difficult objects.

Table 4. Comparison of experimental results before and after using the MTDA strategy.

Scheme	Method	P (%)	R (%)	F_1 (%)	mAP (%)
A	Dynamic Mosaic	80.13	72.92	76.36	79.73
B	Dynamic Mosaic + MTDA	**81.84**	**76.45**	**79.05**	**83.41**

In order to demonstrate the advancement of the MTDA strategy, this paper selects the data augmentation strategy in YOLOv5 as the Baseline Scheme for comparison in our study. The main idea of the Baseline Scheme is to increase the probability of using the mixup algorithm based on the mosaic images, as the main purpose is to provide richer training samples, thereby preventing the network overfitting. In the Baseline Scheme, the probability of using mosaic algorithm is 0.66, and the probability of using mixup algorithm is 0.5. Since the input image contains the original image and the mixed image, it takes 500 epochs for the model to reach convergence, and the rest of the parameters are the same as the first part. The experimental results are shown in Table 5.

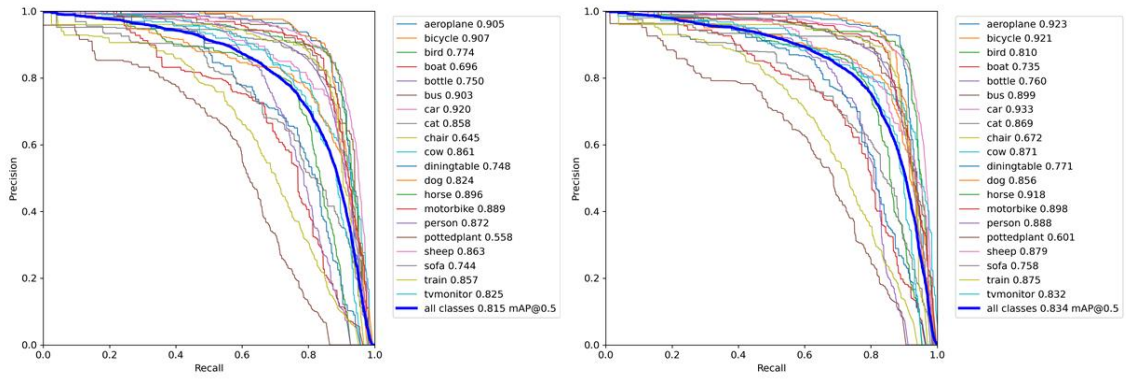
Table 5. Comparison of results of different data augmentation strategies.

Scheme	Method	P (%)	R (%)	F_1 (%)	mAP (%)
A	Baseline	79.43	75.96	77.66	81.54
B	MTDA (Our)	**81.84**	**76.45**	**79.05**	**83.41**

Compared with the Baseline Scheme, the MTDA Scheme has achieved great improvements in various evaluation metrics. Among them, the precision of the model is increased by 2.41%, the recall is increased by 0.49%, the F_1 score is increased by 1.39% and the mAP is increased by 1.87%. This is because the MTDA strategy uses more types of data augmentation methods than the data augmentation strategy in YOLOv5. In addition, the MTDA strategy does not use all data augmentation methods at the same time, but divides the input image into four different parts, and performs different data augmentation processing on each part, improving the information variance between training samples. Therefore, the model can achieve better performance in more complex

detection environments.

In order to further observe the changes of AP values of each class in the Baseline Scheme and the MTDA Scheme, this paper studied the P-R curves of the two schemes, as well as the AP and mAP values of each class, as shown in Figure 9. Compared with the Baseline Scheme, the AP value of all class of the MTDA Scheme has increased. Especially, as the AP value of some classes is low in the Baseline Scheme, the AP value in the MTDA Scheme is greatly improved. For example, the AP value of potted plant increased by 4.3% from 55.8 to 60.1%, and the AP value of boat increased by 3.9% from 69.6 to 73.5%. In addition, the AP values of some classes are already relatively high in the Baseline Scheme, but the AP values are still improved in the MTDA Scheme. For example, the AP value of aeroplane increased by 1.8% from 90.5 to 92.3%, and the AP value of bicycle increased by 1.4% from 90.7 to 92.1%.



(a) (b)

Figure 9. (a) The P-R curve of the Baseline Scheme, and (b) the P-R curve of the MTDA Scheme. The thin curve in the figure represents the P-R curve of each class AP, the thicker blue curve is the P-R curve of mAP, and the corresponding curve color and AP value of each class are given in the rectangle box on the left.

In order to explore the influence of different image combination methods in the MTDA strategy on the recognition accuracy of the model, this paper conducted multiple sets of comparative experiments. The experimental results are shown in Table 6. D_{og} represents the original image, D_{dm} represents the mosaic image generated by the Dynamic Mosaic algorithm, D_{mm} image is superimposed on the basis of the Dynamic Mosaic algorithm and uses the mixup algorithm and D_{mc} image first uses the Dynamic Mosaic algorithm, and then uses the cutout algorithm on this basis. In addition, the proportion of different images in the training samples is the same.

In the process of exploring the influence of different combinations of D_{og}, D_{dm}, D_{mm} and D_{mc} images on the recognition accuracy of the model, we found two interesting phenomena. First, when comparing Schemes A, B, C and D, it can be observed that the trained model has better performance when D_{mm} or D_{mc} images are included in the data augmentation strategy. The reason for this phenomenon is that the D_{mm} and D_{mc} images use the mixup [37] data augmentation algorithm and cutout [30] data augmentation algorithm on the basis of the mosaic images, so that the training samples can generate more transformations, which is important for inducing the robustness of the model. In addition, comparing Schemes B, C and Schemes E, F, we also find that the model containing D_{mc} images have higher recognition accuracy than the model containing D_{mm} images

when other images in the data augmentation strategy are the same. We believe that, compared with the mixup algorithm, the cutout algorithm introduces noise into the image through mask, which can increase the diversity of training samples and make the CNN more robust.

Table 6. Comparison of experimental results of different training sample combinations in MTDA strategy.

Scheme	Training samples				F_1 (%)	mAP (%)
	D_{og}	D_{dm}	D_{mm}	D_{mc}		
A	✓	✓			75.59	78.56
B		✓	✓		77.78	81.31
C		✓		✓	77.68	81.63
D			✓	✓	78.12	82.00
E	✓	✓	✓		77.66	81.54
F	✓	✓		✓	78.07	82.07
G		✓	✓	✓	79.04	83.38
H	✓	✓	✓	✓	**79.05**	**83.41**

Figure 10. Comparison of detection examples of different data augmentation strategies. The detection example comes from the test set of Pascal VOC 2007. It can be seen from the detection examples that the detection model using the Multi-Type Data Augmentation strategy has no missed detection or false detection, and can locate and recognize the target object more accurately.

Figure 10 shows the detection examples of the three different schemes, namely the Original Scheme without data augmentation strategy, Baseline Scheme using data augmentation strategy in YOLOv5, and MTDA Scheme using Multi-Type Data Augmentation strategy. It can be seen from the detection examples that the Original Schemes and Baseline Schemes have problems of false detection and inaccurate positioning. For example, the Original Schemes and Baseline Schemes in the second group of images mistake the bird as a person, and the bounding box of the Original Schemes and Baseline Schemes in the third group of images only encircles the dog's head. In contrast, the MTDA Scheme proposed in this paper has no problems of false detection and inaccurate positioning, and has higher confidence.

4.5. Discussion

In order to summarize the experimental results of this paper and emphasize the advantages of the Dynamic Mosaic algorithm and the MTDA strategy in recognition accuracy, we have compiled the experimental results of the Dynamic Mosaic algorithm and the MTDA strategy, as shown in Table 7.

Table 7. Comparison of Dynamic Mosaic algorithm and MTDA strategy with existing algorithms and strategies.

Scheme	Method	P (%)	R (%)	F_1 (%)	mAP (%)
A	Original	76.03	65.05	70.11	71.19
B	Mosaic [23]	78.99	73.15	75.96	79.01
C	Dynamic Mosaic	80.13	72.92	76.36	79.73
D	Baseline [43]	79.43	75.96	77.66	81.54
E	Dynamic Mosaic + MTDA	**81.84**	**76.45**	**79.05**	**83.41**

By analyzing the above experimental results, we obtained some interesting findings: 1) The Precision of the detection model using the Dynamic Mosaic algorithm has been greatly improved compared with the mosaic algorithm. This shows that the detection model using the Dynamic Mosaic algorithm has better generalization ability, and can reduce the detection error rate of the model for some difficult samples. 2) Compared with the model without the MTDA strategy, the MTDA strategy proposed in this paper can improve the robustness of the model and enable the model to have better detection results when facing complex and difficult objects. Generally, the detection model using the MTDA strategy has better comprehensive performance. 3) The Dynamic Mosaic algorithm and MTDA strategy proposed in this paper only improve the quality of training samples, without changing the structure of the network itself. Therefore, the Dynamic Mosaic algorithm and MTDA strategy will increase the training cost to improve the accuracy of object detection, but will not increase the complexity of the original model.

5. Conclusions

In this paper, we firstly proposed the Dynamic Mosaic algorithm for data augmentation. Based on the original mosaic algorithm, this algorithm adds a dynamic adjustment step, which solves the problem of information waste caused by the large, gray background in the generated mosaic image,

and improves the complexity of the content in the mosaic image. In addition, this paper proposed the Multi-Type Data Augmentation (MTDA) strategy based on the Dynamic Mosaic algorithm to relieve the problem of network overfitting. This strategy divides the training samples into four parts, and uses different data augmentation methods to process the training samples of each part, which improves the information variance between the training samples, and prevents the network from overfitting. Finally, to evaluate the effectiveness of the method, we conducted a series of experiments on the Pascal VOC dataset, and the experimental results show that the Dynamic Mosaic algorithm improves mAP by 0.72% compared with the previous methods, and the MTDA strategy improves mAP by 1.87% compared with other methods. However, we also found that the recognition accuracy of the model for small objects decreased after using the MTDA strategy. This is because the small objects in the training samples will be partially shielded after using the mixup algorithm and the cutout algorithm, which is not conducive to the recognition of small objects. In future work, we hope to solve the problem of decreasing recognition accuracy of small objects by adjusting the ratio of the number of four types of images.

Acknowledgments

This research is jointly supported by the National Natural Science Foundation of China (62072414), and the key scientific and technological project of the Henan Province (212102210104, 222102210071).

Conflict of interest

We declare that we have no conflicts of interest to report regarding this study.

References

1. A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, J. C. W. Lin, G. Fortino, Deep learning for pedestrian collective behavior analysis in smart cities: a model of group trajectory outlier detection, *Inform. Fusion*, **65** (2021), 13–20. https:// doi.org/10.1016/j.inffus.2020.08.003
2. G. Vallathan, A. John, C. Thirumalai, S. K. Mohan, G. Srivastava, J. C. W. Lin, Suspicious activity detection using deep learning in secure assisted living IoT environments, *J. supercomput.*, **77** (2021), 3242–3260. https://doi.org/10.1007/s11227-020-03387-8
3. Y. Djenouri, G. Srivastava, J. C. W. Lin, Fast and accurate convolution neural network for detecting manufacturing data, *IEEE Trans. Ind. Inform.*, **17** (2020), 2947–2955. https://doi.org/10.1109/TII.2020.3001493
4. A. Belhadi, Y. Djenouri, J. C. W. Lin, A. Cano, Trajectory outlier detection: algorithms, taxonomies, evaluation and open challenges, *ACM Trans. Manage. Inform. Syst.*, **11** (2020), 1–29. https://doi.org/10.1145/3399631
5. A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, A. Cano, J. C. W. Lin, A two-phase anomaly detection model for secure intelligent transportation ride-hailing trajectories, *IEEE Trans. Intell. Trans. Syst.*, **22** (2020), 4496–4506. https://doi.org/10.1109/TITS.2020.3022612

6. C. Sun, A. Shrivastava, S. Singh, A. Gupta, Revisiting unreasonable effectiveness of data in deep learning era, in *Proceedings of the IEEE international conference on computer vision*, (2017), 843–852. https://doi.org/10.1109/ICCV.2017.97
7. R. Takahashi, T. Matsubara, K. Uehara, Data augmentation using random image cropping and patching for deep CNNs, *IEEE Trans. Circuits Syst. Video Technol.*, **30** (2019), 2917–2931. https://doi.org/10.1109/TCSVT.2019.2935128
8. C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning (still) requires rethinking generalization, *Commun. ACM*, **64** (2021), 107–115. https://doi.org/10.1145/3446776
9. M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks. in *European conference on computer vision*, (2014), 818–833. https://doi.org/10.1007/978-3-319-10590-1_53
10. L. M. Zintgraf, T. S. Cohen, T. Adel, M. Welling, Visualizing deep neural network decisions: Prediction difference analysis, preprint, arXiv:1702.04595.
11. L. Schmidt, S. Santurka, D. Tsipras, K. Talwar, A. Madry, Adversarially robust generalization requires more data, *Adv. Neural Inform. Process. Syst.*, **31** (2018). https://doi.org/10.48550/arXiv.1804.11285
12. J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, et al., Deep learning scaling is predictable, preprint, arXiv:1712.00409.
13. D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, J. Schmidhuber, Flexible, high performance convolutional neural networks for image classification, in *Twenty-second international joint conference on artificial intelligence*, (2011), 1237–1242.
14. D. Cireşan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, in *IEEE conference on computer vision and pattern recognition*, (2012), 3642–3649. https://doi.org/10.1109/CVPR.2012.6248110
15. C. Shorten, T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, *J. Big Data*, **6** (2019), 1–48. https://doi.org/10.1186/s40537-019-0197-0
16. D. Han, J. Kim, J. Kim, Deep pyramidal residual networks. in *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2017), 5927–5935. https://doi.org/10.1109/ cvpr.2017.668
17. A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Adv. Neural Inform. Process. Syst.*, **6** (2017), 84–90. https://doi.org/10.1145/3065386
18. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, preprint, arXiv:1409.1556.
19. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2016), 770–778.
20. G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2017), 4700–4708
21. S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2017), 1492–1500. https://doi.org/10.1109/CVPR.2017.634

22. Y. Tokozume, Y. Ushiku, T. Harada, Between-class learning for image classification, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2018), 5486–5494. https://doi.org/10.48550/arXiv.1711.10284
23. A. Bochkovskiy, C. Y. Wang, H. Y. M. Liao, Yolov4: Optimal speed and accuracy of object detection, preprint, arXiv:2004.10934.
24. S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in *International conference on machine learning PMLR*, (2015), 448–456.
25. J. Kukačka, V. Golkov, D. Cremers, Regularization for deep learning: A taxonomy, preprint, arXiv:1710.10686.
26. J. Niu, Y. Chen, X. Yu, Z. Li, H. Gao, Data augmentation on defect detection of sanitary ceramics, in *IECON The 46th Annual Conference of the IEEE Industrial Electronics Society*, (2020), 5317–5322. https://doi.org/10.1109/IECON43393.2020.9254518
27. A. Jurio, M. Pagola, M. Galar, C. Lopez-Molina, D. Paternain, A comparison study of different color spaces in clustering based image segmentation, in *International conference on information processing and management of uncertainty in knowledge-based systems*, (2020), 532–541. https://doi.org/10.1007/978-3-642-14058-7_55
28. A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, *Handb. Syst. Autoimmune Dis.*, 2009.
29. F. J. Moreno-Barea, F. Strazzera, J. M. Jerez, D. Urda, L. Franco, Forward noise adjustment scheme for data augmentation, in *IEEE symposium series on computational intelligence (SSCI)*, (2018), 728–734. https://doi.org/10.1109/SSCI.2018.8628917
30. T. DeVries, G. W. Taylor, Improved regularization of convolutional neural networks with cutout, 2017, preprint, arXiv:1708.04552.
31. E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, Q. V. Le, Autoaugment: Learning augmentation policies from data, preprint, arXiv:1805.09501.
32. J. Gui, Z. Sun, Y. Wen, D. Tao, J. Ye, A review on generative adversarial networks: Algorithms, theory, and applications, *IEEE Trans. Knowl. Data Eng.*, 2021. https://doi.org/10.1109/TKDE.2021.3130191
33. D. Ho, E. Liang, X. Chen, I. Stoica, P. Abbeel, Population based augmentation: Efficient learning of augmentation policy schedules, in *International Conference on Machine Learning*, (2019), 2731–2741. https://doi.org/10.48550/arXiv.1905.05393
34. S. Lim, I. Kim, T. Kim, C. Kim, S. Kim, Fast autoaugment, *Adv. Neural Inform. Process. Syst.*, 32 (2019).
35. M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, H. Greenspan, Synthetic data augmentation using GAN for improved liver lesion classification, in *IEEE 15th international symposium on biomedical imaging (ISBI)*, (2018), 289–293. https://doi.org/10.1109/ISBI.2018.8363576
36. A. Raghunathan, S. M. Xie, F. Yang, J. C. Duchi, P. Liang, Adversarial training can hurt generalization, preprint, arXiv:1906.06032.
37. H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, 2017, preprint, arXiv:1710.09412.
38. R. Takahashi, T. Matsubara, K. Uehara, Ricap: Random image cropping and patching data augmentation for deep cnns, in *Asian conference on machine learning*, (2018), 786–798.

39. H. Guo, Y. Mao, R. Zhang, Mixup as locally linear out-of-manifold regularization, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **33** (2019), 3714–3722. https://doi.org/ 10.48550/arXiv.1809.02499
40. S.Yun, D. Han, S. J. Oh, S. Chun, J. Choe, Y. Yoo, Cutmix: Regularization strategy to train strong classifiers with localizable features, in *Proceedings of the IEEE/CVF international conference on computer vision*, (2019), 6023–6032.
41. C. Summers, M. J. Dinneen, Improved mixed-example data augmentation, in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, (2019), 1262–1270.
42. M. Everingham, S. M. Eslami, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, *Int. J. Comput. Vision*, **111** (2015), 98–136. https://doi.org/10.1007/s11263-014-0733-5
43. J. Glenn, S. Alex, B. Jirka, ultralytics/yolov5:v5.0 – YOLOv5 -P6 1280 models, 2021. Available from: https://github.com/ultralytics/yolov5.
44. I. Loshchilov, F. Hutter, Sgdr: Stochastic gradient descent with warm restarts, preprint, arXiv:1608.03983.
45. W. Hao, S. Zhili, Improved mosaic: Algorithms for more complex images, in *Journal of Physics: Conference Series*, **1684** (2020), 012094. https://doi.org/ 10.1088/1742-6596/1684/1/012094

Appendix

In order to make it easier for readers to find and understand the meaning of the symbols in the paper, Table A1 lists each symbol and its meaning in the order in which the symbols appear.

Table A1. Symbols appearing in this paper and their meanings.

Symbols	Meanings
(x_c,y_c)	The center point splicing coordinates of mosaic image
pad_h	The length from the border of spliced images to the border of the mosaic image
pad_w	The width from the border of spliced images to the border of the mosaic image
τ	The threshold for spliced images to borders of mosaic images
(a_i,b_i)	The coordinates of the upper left corner of the respliced image in the mosaic image
(c_i,d_i)	The coordinates of the lower right corner of the respliced image in the mosaic image
h_1	The length of the spliced image
w_1	The width of the spliced image
h_2	The length of the respliced image
w_2	The width of the respliced image
s	The size of the mosaic image
$\max(a, b)$	Taking the maximum value among a and b
$\min(a, b)$	Taking the minimum value among a and b
D_{og}	The original image in the training sample
D_{dm}	The mosaic image generated after the Dynamic Mosaic algorithm
D_{mm}	The mosaic-mixup image generated after the Dynamic Mosaic and mixup algorithm
D_{mc}	The mosaic-cutout image generated after the Dynamic Mosaic and cutout algorithm
D_{out}	The output image after data augmentation
W_1	The ratio of the number of original images to the number of output images

Continued on next page

Symbols	Meanings
W_2	The ratio of the number of mosaic images to the number of output images
W_3	The ratio of the number of mosaic-mixup images to the number of output images
W_4	The ratio of the number of mosaic-cutout images to the number of output images
x_i, x_j	Two images that are subjected to convex linear interpolation in the mixup algorithm
y_i, y_j	The label of the image being performed convex linear interpolation
x	The mixed image generated after the mixup algorithm
y	The label of the mixed image generated after the mixup algorithm
λ	A real number, $\lambda \in [0, 1]$ and conforms to the Beta distribution
β	A real number, $\beta \in (0,\infty)$
P	Representative the evaluation metrics precision
R	Representative the evaluation metrics recall

AIMS Press