Data enhancement method for object detection

Zhijian Yin¹, Lei Xu¹, Hanqing Yu¹, Zhen Yang¹

¹School of Communication and Electronics, Jiangxi Normal University of Science and Technology, Nanchang, Jiangxi, China

*Corresponding author’s e-mail: yangzhen@jxstnu.edu.cn

Abstract—A data enhancement method for object detection is proposed to address the problem that there are few data enhancement means for target detection. This method has obvious advantages for small targets and scenarios where the target background is lacking. It is also a method to solve the problem of small and unbalanced samples. In this paper, the object is subtracted from the bounding box, and after processing, it is fused into a new image using various image classification improvement means such as random brightness, random contrast, random rotation, random cropping, etc. Using mixup-like fusion methods, and XIOU is proposed to guarantee the intersection ratio of the target synthesis during synthesis. An improvement of 2% is obtained on yolov3 over the VOC dataset.

1. Introduction
The fusing map strategy to separate target and background is proposed to improve the performance of target detection. It has been shown to be effective for guidance, especially when there are only a few hundred or thousand records on small datasets with small data samples, such as COCO128 so that the network can be better promoted and provide better object positioning. Train the image by dividing the target away from the entire large graph, by inserting it into other images after transformation, or by increasing the random noise. This approach is not desirable and leads to a decrease in the accuracy of the early training compared to before the enhancement. Therefore, we propose a strategy to improve the map¹; the entire cut target in the training image, where the real ground label is also proportionally mixed with the patch area. By creating rotation changes, transparency changes, stretching, subtly unifying the use of training pixels, and maintaining regional dropouts, CIFAR and image network classification tasks and image network localization tasks with weak supervision are always superior to the most advanced enhancement strategies. In addition, unlike previous enhancement methods, our way of training image network classifiers can achieve consistent performance improvements and MS-COCO image caption benchmarks in Pascal detection when used as a pre-training model². We also show that this approach improves model robustness for input corruption and its out-of-distributed detection performance.

2. Related Works
Regional dropout: A method for removing random areas in an image is proposed to improve the overall performance of CN. Object positioning methods also use regional drop-out techniques to improve the localization of CNs. Our method is similar to these methods, and the key difference is that the removed area is filled with patches from another training image. DropBlock summarizes regional dropouts as functional spaces and shows enhanced replicability³. Our method can also be performed in the functional space.
**Synthesizing training data:** Some work explores integrated training data for further promotion. Create new training samples through the modeling image network to make the model focus more on shape than texture for better classification and object detection performance. Our approach also improves performance on many computer vision tasks by cutting and pasting patches in small batches to generate new samples. Unlike our approach, our approach incurs only negligible additional training costs. For object detection\[4\], the object insertion method is recommended as a background object composition method.

**Mixup:** Our approach is similar to Mixup's in that both samples combine two samples, where the ground authenticity label of the new sample is given by a linear interpolation of a thermal label. As we will see in our experiments, the Mixup samples are affected by local ambiguity and unnaturalness, which confuses the model, especially localization. Recently, hybrid variants have been proposed: they perform feature-level interpolation and other types of transformations. However, the above works often lack an in-depth analysis of localization capabilities and transfer performance\[5\].

**Tricks for training deep networks:** The efficient training of Deep networks is one of the most important issues in the computer vision community because they are very computationally intensive and require a lot of data. Methods such as weight attenuation\[6\], dropout, and batch normalization are widely used for efficient training depth networks. Recently, it has been proposed ways to add noise to the internal features of the CNN or to incorporate additional paths in the architecture to improve image classification performance.

3. **Method**

In each random batch, all targets are deducted. After random enhancement transformation, it is randomly combined with the original image to form a new group of images. Each original image and target is randomly combined with 2 to 5 new targets to form an image. When combining, you need to check the frame to avoid the target being completely covered by the other targets. In addition, you need to ensure a certain coverage ratio, which can enhance the robustness of the model\[7\].

3.1 **XIOU Method**

In processing the training image, the crucial step is to insert the target frame into the original image. Without considering the relationship between the original image and the size of the target image, direct mapping leads to the problem that the target object in the original image will be occluded in a large area if the size of the target object is larger than the size of the target object in the original image. Due to the excessive occlusion, the object in the original image loses the crucial part of the information. The trained model cannot learn the critical features of the object in the original image. Based on this, this experiment proposes a new method to calculate the object overlap degree IOU, called XIOU. This approach considers three relationships between the mapping target and the original target. The specific mapping steps are as follows: First, a central coordinate is randomly generated within the original map\[8\], and the target is moved to the generated central coordinate. Second, a simple coordinate analysis is performed to determine whether there is an overlap between the mapped target and the coordinates of each target on the original map. If there is no overlapping area, it means that every two targets in the map are in a relationship away from each other, then there is no need to calculate XIOU.

When the mapping target and the target in the original map have intersections or overlaps, it is necessary to calculate XIOU according to formula 1. Equation 1 consists of two parts: 1) the first half determines the intersection between the target objects; 2) the second half determines the separation of the two objects. Here is an extreme example, if the two targets overlap and the overlap area is half of the larger target area. In this case, the first half of the equation yields 0.25, and the second half is also 0.25, so the final XIOU is 0.5. We use a value between 0.5 and 1 as a threshold to control the degree of overlap between two bounding boxes, to control the effect of compositing. If the overlapping area of two objects is bigger than half of the larger target area, then the calculated XIOU will be bigger than the threshold value. If the two targets are separated or intersect (not including the case of overlap), then XIOU will
be less than this threshold. Control XIOU can control the target's position in the composite image, the target in the original image will not be covered entirely.

\[
XIOU = \left( \frac{S \cap S_1}{S_1} + \frac{S \cap S_2}{S_2} \right) \ast 0.25 + \left( \frac{S_1}{S} + \frac{S_2}{S} \right) \ast 0.25 \quad (1)
\]

\[
S \cap = S_1 \cap S_2 \quad (2)
\]

\[
0 < XIOU < 1 \quad (3)
\]

Suppose there are two targets in the figure, target 1 and target 2, and their areas are \( S_1 \) and \( S_2 \). The intersection of \( S_1 \) and \( S_2 \) is \( S \). The coordinates of the lower left corner of target 1 are \((x_1, y_1)\), the coordinates of the lower left corner of target 2 are \((x_2, y_2)\), and \( x_1 < x_2, y_1 < y_2 \). The area of the region enclosed by the point \((x_1, y_1)\) and the point \((x_2, y_2)\) is \( S_\Delta \), as shown in the diagonal part of the figure 1.

![XIOU schematic diagram](image)

**Figure 1. XIOU schematic diagram**

### 4. Experimental analysis

#### 4.1 Environment

The hardware configuration for the experiments in this work is Intel (R) Core (TM) i7-8700, the GPU model is NVIDIA GeForce GTX 1050Ti with 8 GB video memory, 8 GB of RAM, and the operating system is ubuntu 16.04. The number of global iterations is 300, the momentum constant \( \beta \) is set to 0.9, the weight attenuation is 0.0005, and the initial learning rate is 0.001. The initial experiment was conducted on yolov5s using COCO128 dataset and VOC dataset, so the experiments in this paper are all conducted on YOLOv5s\[9\].

#### 4.2 Datasets

The VOC2007 and VOC2012 datasets are divided into four broad categories: humans, common animals, traffic vehicles, and interior furniture supplies, which are mainly used for image classification, target detection recognition, and image segmentation\[10\], and the VOC2007 and VOC2012 datasets are mutually exclusive. In this paper, the training set and verification set of VOC2007 and VOC2012 are used as the training set of experiments, and the test set of VOC2007 is used as the test set, of which there are 16551 pictures in the training set and 4952 in the test set.

#### 4.3 Experiments baseline

Training the YOLOv5s model on the VOC dataset without adding any data enhancement strategy\[11\], epoch number is 50, and the obtained map@.5 reached 59%.

#### 4.4 Black pixel filling method

The target of detection is first keyed out of the image. The blank position of the image after keying out the target is filled with black pixels, and then this background image is saved in the background folder\[12\].
The keyed-out detection targets are stored in different folders by category. Randomly select n targets from the detection target folder, where the value of n is ranging from 3 to 5. Then, a background image is randomly selected from the background folder, and the selected targets are pasted onto the background image as new training data, see figure 2. The new data is expanded to the original dataset for model training\(^{[13]}\). The results obtained from the training are shown in Table 1, from which it can be seen that the map@.5 obtained from training on the expanded dataset decreases by 2%. It is presumed that the direct violent keying to fill 0 pixels destroys the correlation between the original target and the background leading to a reduced recognition of a part of the targets that need to be recognized by the background.

| Method                  | Map@ .5  |
|------------------------|----------|
| Unused data enhancement| 59.45%   |
| Black pixel fill       | 57.13%   |

**Table 1. Black pixel filling method result**

4.5 Random noise filling method

The black pixel padding method in the previous experiment led to a decrease in model accuracy. Therefore, this experiment uses random noise to explore the effect of the difference infill on the model performance. First, the targets in the dataset are keyed and placed in different directories according to the target classification. The blank parts of the keyed-off target objects are filled with random noise and saved as images separately\(^{[14]}\). Then, n targets are randomly selected, and the targets are pasted on the ultimately deducted target images to generate new training images. The newly obtained training maps are blended into the original dataset for training, see Figure 3. The trained map@.5 is reduced by 25% from the baseline, see table 2. Based on the experimental phenomenon, we infer that the random noise reaches the bottom layer of the network through the convolutional network layer by layer and affects the network performance. In addition, since the mapping keeps the original size, the excessive difference between the background map and the mapping scale when pasted onto the background map also degrades the training results.

| Method                  | Map@ .5  |
|------------------------|----------|
| Unused data enhancement| 59.45%   |
| Random noise filling    | 26.24%   |

**Table 2. Random noise filling method result**

4.6 Adding targets to the background

The earlier experiments were all about keying the target out of the background and recombining it to compose a new image. Neither the black background fill nor the random noise fill can improve the model accuracy\(^{[15]}\). So this experiment improves the method of synthesizing images. The original image without keying operation and the target are saved separately. When synthesizing, targets are randomly selected and pasted onto the original image with targets, and 2~5 additional targets are added to each image to generate a new training set. Each target undergoes a simple stretching and scales transformation
operation before being pasted onto the original image. When pasting, we follow the principle that the newly added targets do not intersect with the targets in the original image, as shown in figure 4. This strategy also leads to some problems, for example, when the target in the original image occupies a large proportion of the original image. Since the newly added targets cannot intersect with the targets in the original image, then the newly added targets can only be reduced to a relatively small size\cite{16}, leading to more small targets in the generated dataset, which affects the network's performance. The experiment results are in table 3.

| Method                     | Map@.5   |
|----------------------------|----------|
| Unused data enhancement    | 59.45%   |
| Adding targets to the      | 58.52%   |
| background                 |          |

4.7 Coordinate control with XIOU
We introduce the XIOU introduced in the previous method based on Experiment 6.4. This method calculates the IOU for the newly added target and the target in the original image, as shown in equation (1). We mapped with a threshold value less than 0.5. In this way, the targets in the original map can be in three states: away from each other, intersecting, and overlapping. However, it is essential to note that the area of intersection or overlap here will not exceed half the size of the larger scale target\cite{17}. We can control the size of the threshold to control the mapping form, including controlling the maximum occlusion area of the overlapping part\cite{18}. The following figure 5 shows the detail.

![Figure 5. Coordinate control with XIOU](image)

4.8 Mapping target scale transformation
Targets are keyed from the target dataset and stored in category directories, saving the original image and the keyed targets separately. When compositing, the targets are randomly selected and added to the original image\cite{19}, and 2–5 additional targets are added to each image. When adding targets, the previously mentioned XIOU method is introduced to calculate XIOU for the newly added targets and the original targets, with a specified threshold value less than 0.5, and the relative positions of the two targets are controlled by the threshold size. The experiment results are in table 4. The spliced new graph is expanded to the original dataset to generate a new training set\cite{20}, as shown in Figure 6.
Table 4. Coordinate control with XIOU result

| Method                      | Map@.5 |
|-----------------------------|--------|
| Unused data enhancement     | 59.45  |
| Adding targets to the       | 61.97  |
| background                  |        |

5. Conclusion
Most of the current enhancement methods for target detection still use the whole image as the enhancement object. In this paper, based on the idea that most target detection does not require background information, the target and background are separated. The separated target and the background of the keyed target are enhanced separately by transformations such as random brightness, saturation and random cropping. The transformed targets are then stitched back into the original image so that a new target detection dataset is synthesized. This dataset can enhance the generalization ability of the model. The map@.5 of the model after data enhancement improves by xx percentage points over the original model. However, the hyperparameters in this enhancement method still have a great influence on the experimental effect, and the next step will be to further explore the influence of hyperparameters on the model.

ACKNOWLEDGMENTS
This work was supported by the National Natural Science Foundation of China (62061019, 61866016), the Key Laboratory of System Control and Information Processing, Ministry of Education (Scip202106); the Key Project of the Ministry of Education of Jiangxi (GJJ201107) and the General Project of the Ministry of Education of Jiangxi Province (GJJ190587), the Youth Top Project of Jiangxi Science and Technology Normal University (2018QNBJRC002); General Project of the Natural Science Foundation of Jiangxi Province (20202BABL202014).

References
[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4):834–848, 2018.
[2] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2219–2228, 2019.
[3] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout.arXiv preprint arXiv:1708.04552, 2017.
[4] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In Proceedings of the European Conference on Computer Vision (ECCV), pages 364–380, 2018.
[5] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In Proceedings of the IEEE International Conference on Computer Vision, pages 1301–1310, 2017.
[6] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. International Journal of Computer Vision, 88(2):303-338, June 2010.
[7] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness.arXiv preprint arXiv:1811.12231, 2018.
[8] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. In Advances in Neural Information Processing Systems, pages 10750–10760, 2018.

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. InInternational Conference on Learning Representations, 2015.

[10] Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization.arXiv preprint. arXiv:1809.02499, 2018.

[11] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. InCVPR, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren. Deep residual learning for image recognition. InCVPR, 2016.

[13] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. InInternational Conference on Learning Representations, 2017.

[14] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. InAdvances in Neural Information Processing Systems, pages 9423–9433, 2018.

[15] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. InarXiv:1709.01507, 2017.

[16] Gao Huang, Zhuang Liu, and Kilian Q Weinberger. Densely connected convolutional networks. InCVPR, 2017.

[17] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth. InECCV, 2016.

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. InICML, 2015.

[19] Hanjoo Kim, Minkyu Kim, Dongjoo Seo. NSML: meet the mlaas platform with a real-world case study. CoRR, abs/1810.09957, 2018.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. InNIPS, 2012.