2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Data Augmentation via Bit-Plane Manipulation for Object Detection

Changcheng Lu[1,2], Songjie Du[1,2], Weiguo Pan[1,2(✉)], Bingxin Xu[1,2], Nuoya Li[1,2]

[1] Beijing Key Laboratory of Information Service Engineering,
Beijing Union University, Beijing, 100101 China
[2] College of Robotics, Beijing Union University, Beijing, 100101, China
ldtweiguo@buu.edu.cn

Abstract. Current object detection algorithms based on deep learning- heavily depend on a substantial amount of annotated data for model training. High-quality datasets are crucial in addressing challenges such as overfitting. However, collecting large amount of annotated data poses challenging in certain fields. To mitigate this limitation, this paper introduces a data augmentation method based on low-bit plane manipulation. Specifically, this paper employs selected data augmentation methods by processing the low bit planes of the annotated regions in images. This can modify the low-frequency information of the images while minimizing significant visual changes. It is crucial for tasks that depend on high-quality image. During the bit-plane combination process, the augmented image data is achieved through the combination of different bit planes, thereby increasing the diversity of training data. The effectiveness of the proposed method is validated on existing object detection and classification methods, demonstrating notable performance improvements on public datasets, voc2007, voc2012, and kitti2D. These results demonstrating its applicability to object detection and classification that require high-quality input images, enhancing the performance of the algorithms. The code and data can be find here: https://github.com/cjjhf/Data_augmentation.

Keywords: Data Augmentation, Bit-Plane Manipulation, Object Detection.

1 Introduction

In the field of deep learning, various models, such as feedforward neural networks, CNNs (convolutional neural networks) [1], recurrent neural networks, and long short-term memory networks, exhibit inherent advantages in handling high-dimensional data. They possess the capability to automatically extract features and subsequently classify, recognize, predict, or make decisions based on these features [2,3]. Consequently, numerous CNN models have emerged, including LeNet-5 [4], VGG-16 [5], AlexNet [1], ResNet [6], GoogLeNet [7]. Currently, the prevailing trend approach involves attempts to compensate for data insufficiency by enhancing model capabilities. Collecting high-quality datasets demands substantial financial and logistical investments, especially in the field of medicine [8]. To obtain more diverse data, the current trend in training introduces the concept of data augmentation. There are solutions aimed at improving object detection accuracy, including those based on the transformer mechanism [9], those generating optimal network structures based on NAS [10], and those employing multimodal approaches [11]. Recently, research on weakly labeled images [12] and weakly supervised object detection [13] has become a hot topic.

Mixed transformations encompass various techniques, including mixup [14,15], cutmix [16], cutout [17], and others. These methods are designed to effectively mitigate the model's tendency to memorize incorrect labels. By prompting the model to recognize objects from a local perspective and integrating information from other samples into the cropped region. The modified regions may lead to information loss from the original samples, making them unsuitable for all tasks and scenarios. In contrast, generative data augmentation methods, such as Variational Autoencoders (VAE) [18], Generative Adversarial Networks (GAN) [19], and Diffusion models [20], involve the generation of new images to augment data diversity, representing an innovative development direction. Image style transfer [21] is a frequently employed augmentation method, where models are initially trained using a similar high-quality dataset and subsequently fine-tuned them with original dataset. However, it is may heighten the model's uncertainty [22] in perceiving real-world scenarios. AutoAugment [23] addresses the challenge of determining the best augmentation strategy by framing it as a discrete search problem [24]. It directly searches for the optimal strategy tailored to a specific dataset, the computational cost is considerable. In [25], the authors propose a random augmentation method. Unlike AutoAugment, it avoids using specific probabilities to decide whether to employ a particular sub-strategy. In [26], it can regenerated urban layout for the target region. GridMask [27] generates a mask with the same resolution as the original image, randomly flips the mask, and multiplies it with the original image to obtain the augmented image. The size of the generated mask grid is controlled by hyperparameters. Nevertheless, concerns arise due to the visually perceived change in the augmented image. Color alteration is also a frequently used augmentation technique. In [28], it proposes a high-quality fully-automatic colorization method using deep learning. Color space transformation involves adjusting the brightness deviation within the dataset to enhance the model's adaptability to different lighting conditions.

In the medical and autonomous driving fields, the shortage of high-quality datasets presents a significant challenge. While data augmentation play a crucial role in addressing this issue, conventional methods are not be suitable in these fields.

The aforementioned data augmentation methods often introduce visible changes to the images, and some methods may significantly impact the quality, leading to the loss of valuable information. The data augmentation method proposed in this paper operates on the bit planes of the image. The processed method exhibit minimal visual changes, avoiding significant degradation in image quality. The contributions of this paper are summarized as follows:

(1) To increase the diversity of the training data, we propose a bit-plane based augmentation method by applying selected data augmentation methods within the annotated regions of the low bit planes, it is effectively modify the visual content of the images without introducing significant disruptions.

(2) During the bit-plane recombination stage, reconstruction is performed by modifying different bit-plane sequences, which leads to the generation of final augmented images. This process is designed to enhance the robustness of the training model.

2 Propose method

In this paper, we propose an augmentation method designed for manipulating the low-bit planes of the images. This approach results in augmented image data that has minimal visual difference from the original data but can enhance the performance of object detection algorithms. As illustrated in the Fig. 1 the proposed method can be divided into four steps: bit-plane decomposition, bit-plane data augmentation, bit-plane processing, and bit-plane composition.

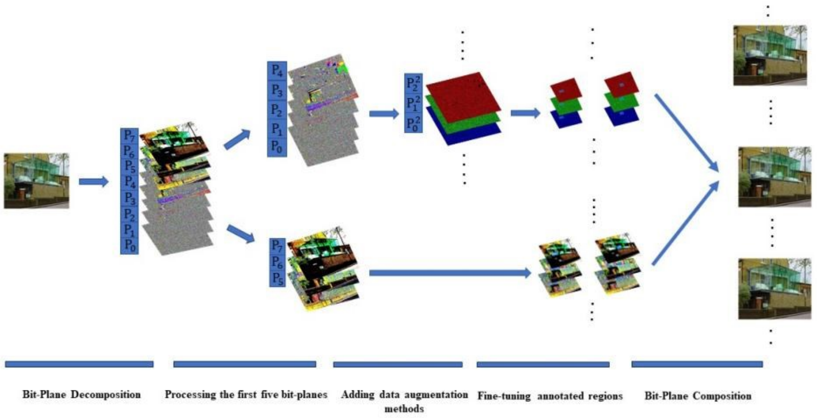

P7
P6
P5
P4
P3
P2
P1
P0
P4
P3
P2
P1
P0
P2^2
P1^2
P0^2
P7
P6
P5
Bit-Plane Decomposition
Processing the first five bit-planes
Adding data augmentation methods
Fine-tuning annotated regions
Bit-Plane Composition

Fig. 1. Proposed Data Augmentation Framework.

A. Bit-Plane Decompositions

A pixel in an image can be represented by an 8-bit binary value, ranging from 0 to 255 as shown in the following:

$$\begin{aligned} value = {} & a_7 \times 2^7 + a_6 \times 2^6 + a_5 \times 2^5 + a_4 \times 2^4 \\ & + a_3 \times 2^3 + a_2 \times 2^2 + a_1 \times 2^1 + a_0 \times 2^0 \end{aligned} \tag{1}$$

here $value$ represents the pixel value. a_i represents the weights, each of which varies, with a_7 having the highest weight and a_0 having the lowest. This implies that the value of a_7 has the most significant impact on the image, while the value of a_0 has the least, as illustrated in Fig. 2. By exploiting this characteristic during the processing of bit planes, focusing on the lower bit planes leads to visually subtle differences in the generated images.

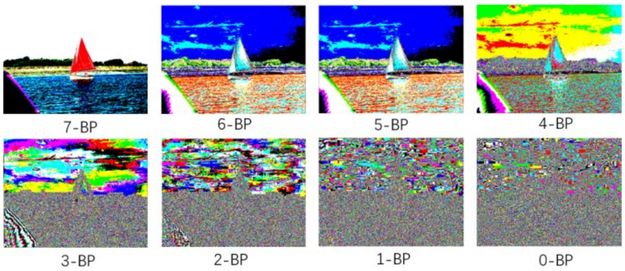


Fig. 2. visualization of different bit planes.

The internal structural information of each extracted bit plane is distinct, with lower planes containing fewer internal structural details. An extraction matrix is constructed to extract bit planes from the image. To ensure extraction from a color image, the extraction process is performed separately for each channel of every bit plane. The formula is as follows:

$$\begin{gathered} p_{c,ori}^{m} = p_{ori} * 2^{m} * n_{c} \\ m = \left[0, 7\right], c = \left[0, 2\right] \end{gathered} \tag{2}$$

$p_{c,ori}^{m}$ represents the extracted image, specifically in terms of channels. p_{ori} is the original image. 2^{m} and n_{c} are extraction matrices, where 2^{m} extracts a specific bit plane, and n_{c} extracts a particular channel. This process ultimately yields the desired information.

To facilitate subsequent operations, individual channels of every bit plane are partitioned, and subsequently consolidated, as depicted by the following formula:

$$p_{pro} = \sum_{m=1}^{7}\sum_{c=0}^{2} p_{c,ori}^{m} \tag{3}$$

p_{pro} denotes the sum of the split images, where $p_{c,ori}^{m}$ denotes the bit-plane image obtained by decomposing the original image.

B. Data augmentation on bit planes

To maximize the utilization of expandable space in the lower bit planes, data augmentation is applied to the annotated regions of the lower planes. The goal is to enhance data diversity and robustness, while minimizing visually perceptible alterations in the augmented images. Augmentation methods employed in this study include flipping, blurring, noise injection, cutout, sharpening, highlighting, and color jittering. These methods perturb annotated regions in target images by manipulating spatial positions, adding interference information, random cropping, colors alteration, and modifying details and high-frequency information. The specific parameters of the aforementioned method are shown in the table below.

Table 1. Parameters of the data augmentation

Method	Parameters
flip	left-right flipping
blur	Gaussian blur kernel 99×99
noise injection	The Poisson distribution λ is 2
cutout	covered area:16% - 50%
sharpen	kernel [0, -1, 0; -1, 2, -1; 0, -1, 0]
highlight	The brightness increases by 18%
color jitter	The color shift is 15

C. Bit-Plane Processing

In the subsequent steps, processing will be conducted on each channel of every bit plane. The specific procedure involves data augmentation operations on the annotated regions of the lower bit planes, as delineated by the formula:

$$P_{c,pro}^{m} = \sum_{m=1}^{i}\sum_{c=0}^{2}\left(p_{c,ori}^{m} + n\right) \qquad i = \left[0, 7\right] \tag{4}$$

here $p_{c,pro}^{m}$ represents the sum of the processed channel images, $p_{c,ori}^{m}$ is the bit-plane image obtained by decomposing the original image, and n denotes the data augmentation method. The resulting image exhibits minimal visual changes, while alterations have occurred in the low-frequency information of the image.

While introducing noise may accentuate discrepancies between the original image and the noise-added image version, it could also potentially impede the model's ability to extract meaningful features from the data, thereby increasing the risk of overfitting or underfitting. To address this concern, we employ image quality evaluation and mAP as key criteria for determining the appropriate bit planes for noise addition. For image quality evaluation, we utilize PSNR[29], MS-SSIM[30], and VIF [31] metrics. These metrics serve as criteria for selecting the appropriate bit planes for noise addition, taking into account their effects on image quality and subsequent model performance. Fig. 3 and Table 2 show some data augmentation results. For first image, without augmentation, the PSNR is 32.4, indicating high quality. After flip augmentation, the quality slightly decreases (PSNR 31.7), while blur augmentation has a greater impact (PSNR 30.1). MS-SSIM and VIF slightly drop, with flip augmentation having the least effect, while blur augmentation significantly affects visual quality. For image 2, without augmentation, the PSNR is 33.1, showing excellent quality. Noise augmentation reduces the quality (PSNR 30.9), and crop augmentation has the greatest impact (PSNR 29.8). MS-SSIM and VIF both decrease, with crop augmentation causing significant loss of visual information.

Fig. 3. test images.

When PSNR exceeds 30dB, it becomes challenging for the human eye to perceive differences between the compressed and original images. Within the 10dB to 20dB range, the original structure of the image remains perceptible, with minimal visual distinctions between the two images. This study selects images with PSNR values surpassing 30dB as suitable candidates. As indicated in Fig. 4, the number of processed bit planes increases, images quality gradually diminishes. Processing the initial five bit planes maintains PSNR values above 30dB, MS-SSIM values hovering around 0.99, and VIF values show a gradual decline. However, mAP exhibits a different trend, initially decreasing over the first four bit planes but subsequently increasing with the fifth bit plane. After considering all factors, opting for the first five bit planes (i.e., bit planes 0, 1, 2, 3, and 4) emerges as the most optimal approach. This selection ensuring both high PSNR values and the preservation of overall visual coherence, as demonstrated in Fig. 5.

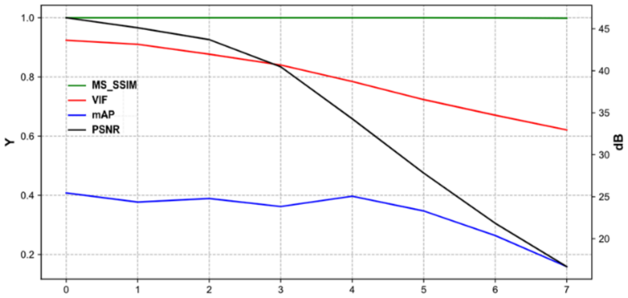


Fig. 4. Comparison of Image Quality Metrics.

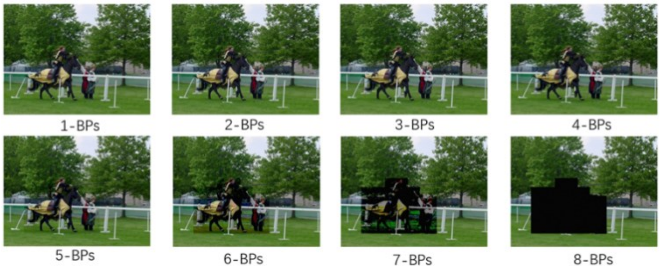
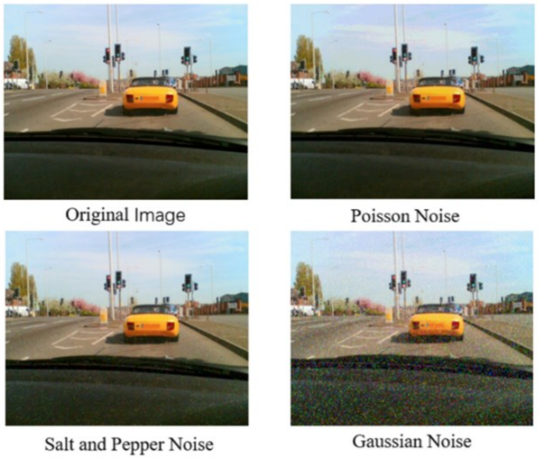


Fig. 5. Different Bit-Plane Processing

The criteria for selecting noise entail minimal visual impact on the image while induces internal changes. Firstly, Gaussian noise is considered, generated by incorporating random values from a normal distribution with a mean of zero and a specified standard deviation into the input data. Salt-and-pepper noise refers to two types of noise: salt noise and pepper noise. Salt noise typically manifests white noise, while pepper noise is typically characterized as black noise. The former denotes high-intensity noise, while the latter pertains to low-intensity noise. When both types of noise are present, they manifest as black and white specks within the image. As shown in Fig.6, under the identical conditions, among these four types of noise, Poisson noise exhibits the least impact.


Original Image

Poisson Noise

Salt and Pepper Noise

Gaussian Noise

Fig. 6. Comparison of Different Noises

Fine-tuning the annotated regions involves processing multiple annotated regions in a single image, with all regions adjusted in each iteration. The fine-tuning process includes making small adjustments to the bounding boxes in all nine directions, including no movement. The purpose of this movement is to provide additional contextual information, assisting the model in understanding the position and relationships of the objects within the overall scene. Such measures serve to mitigate overfitting and enhances the model's generalization capabilities.

Table 2. Different data augmentation methods applied to two images

Image ID	Data Augmentation Method	PSNR	MS-SSIM	VIF
1	No	32.4	0.987	0.91
1	Flip	31.7	0.985	0.89
1	Blur	30.1	0.980	0.87
2	No	33.1	0.990	0.93
2	Noise	30.9	0.982	0.88
2	Crop	29.8	0.976	0.85

D. ***Bit-plane combination***

In this step, the low bit planes are integrated with the high bit planes using three methods: sequential, interleaved, and deletion.

Given that the internal structural information in an image follows a sequential arrangement, the combination proceeds from the low bit planes to the high bit planes. In conventional combination methods, the discussion specifically focuses on which bit planes to process. This can be formulated by the formula:

$$P_{combin} = \sum_{m=0}^{i}\sum_{c=0}^{2}(p_{c,ori}^{m} + n) + \sum_{m=i}^{7}\sum_{c=0}^{2} b_{c,ori}^{m} \quad i = \left[0, 7\right] \tag{5}$$

P_{combin} represents the aggregated image after adding noise, $p_{c,ori}^{m}$ is the bit-plane image obtained by decomposing the original image, n is a fixed method, and $b_{c,ori}^{m}$ is the image without the applied method. This approach aligns with the internal structure of the image, maximizing information preservation and maintaining image integrity.

To provide a clearer depiction of the combination method that best aligns with the expected results, an experiment was conducted using 6000 images from the VOC2007 dataset [32] for validation. The experiment comprised two distinct groups: the original and the augmented dataset. The augmented group was generated through the inclusion of noise addition as a representative example.

The experiment is divided into two groups: interleaving and deletion. The regular group serves as the control. In the deletion experiment group, individual bit planes from the first to the fifth are systematically deleted. Conversely, in the interleaving experiment group, bit planes spanning from the first to the fifth are interwoven with each other, yielding a total of 10 combination methods.

The experimental platform utilized in this study is OpenMMLab [33], version 2.0, which serves as the platform for implementing the training of a Faster R-CNN model [34] with ResNet-50 [35] as the backbone. The learning rate is set to 0.0025, and the experiment is trained for 24 epochs. Feature Pyramid Network (FPN) [36] is also employed, which enhances the model's capability to perceive objects across various scales by providing multi-scale feature maps. In the domain of object detection, FPN plays a pivotal role in furnishing superior contextual information and extracting detailed features, thereby contributing to the heightened accuracy and stability of detection results.

Table 3. Training Results for Specific Bit Plane Deletions.

Category	0	1	2	3	4
all-com	0.7095	0.7043	0.6966	0.6905	0.6937
all-pro	0.7088	0.7043	0.7069	0.702	0.6804
part-com	0.7194	0.7070	0.7129	0.6964	0.6941
part-pro	0.7204	0.7150	0.7194	0.7154	0.6865

Table 3 presents the results of removing specific bit planes. Denoted as planes 0 through 4, with one plane removed per iteration. "all-com" indicates the overall mAP result, "all-pro" denotes the mAP after augmentation, "part-com" represents the mAP of the original dataset, and "part-pro" represents the mAP of the augmented dataset. The original dataset achieves an object detection accuracy of 0.7134. As indicated in the table, there is a trend of decreasing overall precision with the increasing index of the deleted bit plane. Notably, a minor improvement is observed only when deleting the third bit plane, yielding a precision of 0.7129 for the original processing and 0.7194 for the augmented processing. However, this increment does not significantly affect the overall declining trend. Particularly noteworthy in the case of the fifth bit plane, the results after processing exhibit inferior compared to the original processing, with noticeable visual changes observed in images processed up to the fifth bit plane.

Considering the comprehensive dataset, it becomes evident that as the index of the deleted bit plane increases, precision decreases, occasionally dropping below the precision of the original untreated data. Images processed with augmentation yields higher precision compared to non-augmented processing, though a shift in trend is noticeable starting from the fifth bit plane. Notably, for the fifth bit plane, the precision of the original processing is surpasses than that of the processed images.

Based the experimental data, it is apparent that the detection accuracy achieved in the experiment is lower than the detection accuracy of the original data, failing to meet the selection criteria. Nevertheless, several conclusions can still be drawn from the experimental data.

When low bit planes are interleaved with high bit planes, there is an observable increase in the overall object detection accuracy. However, if there exists a significant difference in bit plane index, the object detection accuracy experiences a subsequent decrease. This phenomenon may be attributed to the significant divergence in information carried by low bit planes when interleaved with high bit planes, thereby resulting in a decline in object detection accuracy.

3 Results and Discussion

A. ***Model and Experiment Settings***

This study employed both single-stage object detection models, YOLOv7 [37] and YOLOv8, alongside two-stage detection models including Faster R-CNN, Cascade R-CNN [38], and RetinaNet [39], as well as object detection models DiffusionDet [40] and EfficientDet[41], and image classification models EfficientNetV2 [42], ResNet 50, and Vision Transformer [43].The single-stage object detection models were implemented using the MMDetection framework, while the two-stage object detection models were built upon the PyTorch framework. For the single-stage models, the experiment was configured with 200 iterations, a batch size of 8, and a learning rate of 0.01. The two-stage models were trained with 24 iterations, a batch size of 16, and a learning rate of 0.00125. Training was conducted on a single NVIDIA GeForce RTX 2080 Ti, equipped with a CPU model of Platinum 8352V, PyTorch version 1.11, and CUDA version 11.3.

B. ***Dataset***

The experiment utilized three datasets: VOC2007, VOC2012, and the Kitti2D dataset [38]. The VOC format serves as a standardized dataset format for computer vision tasks, primarily geared towards object detection, image classification, and semantic segmentation. Comprising 20 classes, the dataset includes annotated objects ranging from people and animals (e.g., cats, dogs, birds) to vehicles (e.g., cars, ships, airplanes) and furniture (e.g., chairs, tables, sofas). On average, each image contains 2.4 annotated objects. The VOC2007 dataset comprises 9963 annotated images, while its upgraded counterpart, VOC2012, boasts a total of 17125 images. On the other hand, the Kitti2D dataset contains 7481 annotated images, featuring common road objects such as cars, pedestrians, and cyclists. Primarily associated with autonomous driving, this dataset was tailored for this experiment, focusing on six traffic-related classes. The images are sourced from diverse real-world scenarios, spanning urban and rural environments, rendering it a fitting choice for research in autonomous driving.

Data augmentation was employed across all three datasets, resulting in an augmented dataset utilized for experimentation. To enable direct comparison between the original and augmented datasets, one image was randomly selected from the nine augmented images, making the number of images in the augmented dataset equivalent to the original dataset. The augmented dataset was partitioned into training, validation, and test sets, maintaining a ratio of 7:2:1.

C. ***Evaluation metrics***

Mean Average Precision (mAP) is a performance evaluation metrics within object detection tasks. It combines the accuracy of the detection model across different categories and its robustness to different confidence threshold values, providing a comprehensive performance measure. In this study, mAP@0.5 is utilized for single-stage object detection models, whereas mAP is used for two-stage object detection models. The formula for computing mAP is outlined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$Precision$ represents accuracy, and $Recall$ signifies the recall rate. Here, TP denotes true positives (correctly identified positive instances), FP is false positives (incorrectly identified negative instances as positive), FN stands for false negatives (incorrectly identified positive instances as negative).

AP (Average Precision) serves as a balanced evaluation of precision and recall for each class. The formula for AP is as follows:

$$AP = \int_0^1 P(r)dr \tag{8}$$

mAP (mean Average Precision) is the average precision across all predicted objects for various classes. The formula is as follows:

$$mAP = \frac{1}{k}\sum_{i=1}^{k} AP_i \tag{9}$$

here k is the number of classes, and AP_i represents the average precision for each class.

D. Validation of the Dataset in Image Classification

In this paper, the performance gap between EfficientNetV2 and Vision Transformer varies across different datasets. According to Table 4, in the original dataset, the Top-1 accuracy of EfficientNetV2 is 55.19, while Vision Transformer only reaches 35.58, resulting in a gap of nearly 20%. However, in the dataset used in this paper, EfficientNetV2's accuracy is 54.13, and Vision Transformer improves to 32.82, narrowing the gap to 21.31%, indicating that this dataset provides a more balanced evaluation across different models.

Table 4. Comparison of Image Classification Models on Datasets

Data	Methods	Top-1	Top-5
Origiral data	EfficientNetV2	55.19	94.49
	ResNet 50	40.72	85.91
	Vision Transformer	35.58	81.02
Augmentation data	EfficientNetV2	74.62	97.01
	ResNet50	43.39	88.91
	VisionTransformer	45.66	84.32

The Precision and Recall metrics show that in the original dataset, EfficientNetV2 achieves a Precision of 61 and a Recall of 52.7, while Vision Transformer’s Precision and Recall are much lower, at 30.31 and 17.49, respectively. In the dataset from this paper, Vision Transformer’s Precision improves to 32.38 and Recall to 19.46. Although the absolute values are still low, the improvement suggests that the dataset contributes to better model performance.

This dataset may include more diverse and complex samples, forcing the models to learn more comprehensive global features instead of relying on specific characteristics for predictions. This results in more balanced evaluations between models, highlighting the strengths and weaknesses of each architecture. In conclusion, the dataset in this paper enhances the balanced performance of models, better reflecting their generalization capabilities in complex scenarios and making it suitable for comprehensive evaluation of different model performances.

E. Accuracy of the Model Across Different Categories

The evaluation metric employed in this study is mAP (mean Average Precision). Firstly, an analysis of the average precision for each category is conducted on the VOC2007 dataset. The dataset comprises 20 categories, including "Aeroplane" (AE), "Bicycle" (BI), "Bird" (BR), "Boat" (BO), "Bottle" (BT), "Bus" (BU), "Car" (CA), "Cat" (AT), "Chair" (CH), "Cow" (CO), "Dining Table" (DI), "Dog" (DO), "Horse" (HO), "Motorbike" (MO), "Person" (PR), "Potted Plant" (PO), "Sheep" (SH), "Sofa" (SO), "Train" (TR), and "TV" (TV).The detection models utilized in this study include Faster R-CNN, Cascade R-CNN, RetinaNet, YOLOv7, and YOLOv8, as outlined in Table 5.

Table 5. Comparison of the training results for each category before and after augmentation on the VOC2007 dataset for various models.

	Faster r-cnn	Cascade r-cnn	Retinanet	Yolo v7	Yolo v8	Faster r-cnn	Cascade-rcnn	retinanet	Yolo v7	Yolo v8
mAP	0.74	0.715	0.735	0.872	0.785	0.77	0.752	0.753	**0.91**	0.827
AE	0.788	0.784	0.756	0.922	0.834	0.812	0.804	0.795	0.907	**0.924**
BI	0.808	0.798	0.794	0.917	0.889	0.87	0.868	0.857	**0.943**	0.863
BR	0.765	0.698	0.752	0.9	0.675	0.784	0.715	0.764	**0.979**	0.783
BO	0.57	0.588	0.601	0.753	0.704	0.7	0.687	0.673	**0.902**	0.761
BT	0.559	0.496	0.611	0.735	0.67	0.571	0.591	0.567	**0.81**	0.692
BU	0.775	0.795	0.783	0.943	0.764	0.873	0.893	0.85	**0.96**	0.856
CA	0.802	0.804	0.846	0.602	0.878	0.812	0.812	0.862	0.828	**0.89**
AT	0.867	0.802	0.832	0.856	0.813	0.866	0.865	0.835	**0.947**	0.863
CH	0.573	0.504	0.56	**0.989**	0.599	0.638	0.575	0.616	0.975	0.702
CO	0.775	0.761	0.773	0.886	0.873	0.75	0.744	0.724	**0.921**	0.785
DI	0.704	0.629	0.681	**0.934**	0.764	0.783	0.723	0.789	0.805	0.743
DO	0.783	0.774	0.773	0.945	0.863	0.846	0.782	0.799	**0.974**	0.809
HO	0.896	0.884	0.847	**0.984**	0.863	0.797	0.788	0.812	0.971	0.905
MO	0.793	0.776	0.777	0.955	0.795	0.773	0.788	0.778	**0.983**	0.876
PR	0.795	0.792	0.788	0.922	0.865	0.795	0.796	0.79	**0.973**	0.866
PO	0.477	0.478	0.491	0.833	0.547	0.637	0.665	0.669	**0.909**	0.768
SH	0.785	0.699	0.742	0.938	0.779	0.747	0.676	0.66	**0.974**	0.879
SO	0.701	0.644	0.689	**0.804**	0.756	0.717	0.682	0.666	0.79	0.745
TR	0.79	0.868	0.834	0.811	0.857	0.838	0.798	0.796	0.91	**0.953**
TV	0.788	0.718	0.772	0.81	**0.92**	0.786	0.794	0.751	0.744	0.881

The training results for both the original and augmented datasets are provided separately in the table. It is evident from the table that both single-stage and two-stage detection models show improvements on the augmented dataset. To further emphasize the advantages of the augmented dataset, the achieved results for each category are bolded in the table. Notably, for many categories, the best results are obtained from the augmented dataset. Comparing the training on the original dataset with YOLOv7 and the augmented dataset with YOLOv7, overall, the augmented dataset exhibits higher category accuracy. For instance, in the "Car" (CA) category, the highest accuracy is achieved with augmented YOLOv7. However, when comparing the results before and after augmentation, the model's accuracy is generally higher after augmentation.

F. The generalization capability of the method

To illustrate the feasibility of our proposed method, we conducted testing and validation on the VOC2007, VOC2012, and KITTI datasets. The utilization of both VOC2007 and VOC2012 datasets aimed to evaluate the method’s performance across general datasets, while the KITTI dataset served to evaluate the method's feasibility in specific scenarios. We observed that under comparable dataset conditions, different object detection models exhibited varying improvements in detection accuracy. This indicates that the method proposed in this paper exhibits strong generalization capabilities and has the potential to be extended for handling diverse and previously unseen data.

Table 6. Detection Results on VOC2007 (%)

Method	Original	Augmented	Increase
Faster R-CNN	74.0	77.0	3.0
Cascade R-CNN	71.5	75.2	3.7
Retinanet	73.5	75.3	1.8
Yolo v7	87.2	91	3.8
Yolo v8	78.5	82.7	4.2
DiffusionDet	52.5	57.5	5.0
EfficientDet	73.0	79.5	6.5

In the general dataset, our method demonstrates improvements in both single-stage and two-stage approaches as Table 6 shows. Notably, when comparing single-stage and two-stage object detection models, the degree of enhancement is more pronounced in the one-stage models. Specifically, YOLOv7 exhibits a 3.8% improvement, while YOLOv8 shows a 4.2% enhancement. Similarly, EfficientDet achieves a 6.5% improvement. Among the two-stage models, Faster R-CNN, Cascade R-CNN, and RetinaNet achieved improvements of 3%, 3.7%, and 1.8%, respectively. Moreover, DiffusionDet shows a notable 5.0% enhancement. This comparison highlights the effectiveness of our method across different object detection frameworks.

Table 7. Detection Results on VOC2012 (%)

Method	Original	Augmented	Increase
Faster R-CNN	61.4	71	9.6
Cascade R-CNN	60.1	69.4	9.3
Retinanet	62.9	70.7	7.9
Yolo v7	87.1	93.8	6.7
Yolo v8	71	80.4	9.4
DiffusionDet	55.3	60.0	4.7
EfficientDet	72.5	76.9	4.4

In the general dataset VOC2012, the observed improvements are more pronounced, showing varying degrees of enhancement across different object detection models as Table 7 shows. Notably, Faster R-CNN exhibits the largest increase in accuracy, with a notable improvement of 9.6%. Comparing the training results on VOC2012 with those on VOC2007, it is evident that the improvements are generally superior, surpassing them by 3% to 6%. Specifically, among the models, Faster R-CNN demonstrates the highest improvement, with a boost of 9.6%. In the two-stage models, Cascade R-CNN and RetinaNet achieved improvements of 9.3% and 7.9%, respectively. In the single-stage models, YOLOv7 and YOLOv8 demonstrated improvements of 6.7% and 9.4%. Additionally, EfficientDet achieved a 4.4% improvement, while DiffusionDet demonstrated a 4.7% increase. This highlights the effectiveness of our method across different object detection frameworks in VOC2012.

Table 8. Detection Results on KITTI 2D (%)

Method	Original	Augmented	Increase
Faster R-CNN	83.5	86.6	3.1
Cascade R-CNN	82.7	86.8	4.1
Retinanet	81.9	85.2	3.3
Yolo v7	92.3	96.2	3.9
Yolo v8	84.5	87.2	2.7
DiffusionDet	68.3	72.2	3.9
EfficientDet	37.6	41.0	3.4

On the KITTI 2D dataset tailored for autonomous driving scenarios, our method exhibits notable improvements as Table 8 shows. Among the two-stage models evaluated, the Cascade R-CNN model stands out, showing an improvement of 4.1%. Similarly, within the single-stage models, YOLOv8 emerges as the top performer with a commendable improvement of 3.9%. DiffusionDet demonstrated an improvement of 3.9%, while EfficientDet showed a 3.4% increase. These results signify the positive impact of our method on datasets specifically designed for domains, such as autonomous driving.

The experimental results presented in Table 9 demonstrate the performance of various data augmentation methods when integrated with the YOLOv8 framework, evaluated on the VOC2007 dataset. The mean Average Precision (mAP) metric is used to assess the effectiveness of each augmentation technique. Among the methods tested, the proposed method achieves the highest mAP of 0.827, indicating its superior ability to enhance model performance compared to other augmentation strategies. Notably, traditional methods such as mixup and flipud show moderate improvements, with mAP values of 0.797 and 0.773, respectively. However, techniques like keepaugment result in a relatively lower mAP of 0.765, suggesting limited effectiveness in this context.

Table 9. Different augmentation method comparison

Method	**mAP**
Yolo v8+original	0.785
Yolo v8+mixup	0.797
Yolo v8+flipud	0.773
Yolo v8+keepaugment	0.765
Yolo v8+ours	0.827

4 Conclusion

In addressing the scarcity of high-quality datasets, the proposed method in this paper has demonstrated effectiveness, which lies in integrating bit-plane manipulation with data augmentation, selectively processing different bit planes (e.g., low bit planes) to effectively enhance both detail and overall image information while maintaining visual quality. By applying various data augmentation techniques (such as blurring and noise injection) on low bit planes and fine-tuning the enhanced data during the bit-plane recombination stage, the robustness and accuracy of the detection model are significantly improved. Additionally, the proposed method can be integrated with other models. Future research could further explore how to combine bit-plane decomposition with cross-domain learning to better handle differences between various data domains, and extend this method to other computer vision tasks such as image segmentation and video analysis.

Acknowledgments. This work was supported by the Beijing Natural Science Foundation (4232026, 4242020), National Natural Science Foundation of China (grant nos. 62272049, 62171042, 61871039, 62102033, and 62006020), Scientific Research Projects of Beijing Union University (grant nos. ZK10202202, ZK10202404, BPHR2020DZ02, ZK40202101, and ZK120202104).

Disclosure of Interests. The authors declare no conflict of interest.

References

[1]. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(06), pp:84-90 (2012).

[2]. Ling Dai, Bin Sheng, Tingli Chen, et al. A deep learning system for predicting time to progression of diabetic retinopathy. Nat Med 30, 584–594 (2024).

[3]. Ling Dai, Liang Wu, Huating Li, et al. A Deep Learning System for Detecting Diabetic Retinopathy Across the Disease Spectrum, Nature Communications, 12, 3242 (2021).

[4]. LeCun Y, Bottou L, Bengio Y, P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), pp: 2278-2324 (1998).

[5]. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[6]. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition, IEEE conference on computer vision and pattern recognition. pp: 770-778, IEEE, Las Vegas, NV(2016).

[7]. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions, IEEE conference on computer vision and pattern recognition, pp: 1-9. Boston, MA(2015).

[8]. Bo Qian, Hao Chen , Xiangning Wang, et al. DRAC 2022: A public benchmark for diabetic retinopathy analysis on ultra-wide optical coherence tomography angiography images, Patterns, 5(3), pp:1-10 (2024).

[9]. Xiao Lin, Shuzhou Sun, Wei Huang, et al. EAPT: Efficient Attention Pyramid Transformer for Image Processing. IEEE Trans. Multim. 25: 50-61 (2023).

[10]. Zhihua Chen, Guhao Qiu, Ping Li, et al. MNGNAS: Distilling Adaptive Combination of Multiple Searched Networks for One-Shot Neural Architecture Search. IEEE Trans. Pattern Anal. Mach. Intell. 45(11): 13489-13508 (2023).

[11]. Jiajia Li, Jie Chen, Bin Sheng, et al. Automatic Detection and Classification System of Domestic Waste via Multimodel Cascaded Convolutional Neural Network. IEEE Trans. Ind. Informatics 18(1): 163-173 (2022).

[12]. Yunqiu Xu, Chunluan Zhou, Xin Yu, Yi Yang, Cyclic Self-Training With Proposal Weight

Modulation for Cross-Supervised Object Detection, IEEE Transactions on Image Processing, 32, pp. 1992-2002(2023).

[13]. Yunqiu Xu, Yifan Sun, Zongxin Yang, Jiaxu Miao, Yi Yang, H2FA R-CNN: Holistic and Hierarchical Feature Alignment for Cross-domain Weakly Supervised Object Detection, IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14309-14319, New Orleans, LA, USA(2022).

[14]. Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, David Lopez-Paz, mixup: Beyond empirical risk minimization, International Conference on Learning Representations, pp:1-13,Vancouver Canada(2018).

[15]. Wentao He, Jianfeng Ren, Ruibin Bai, Data augmentation by morphological mixup for solving Raven' s progressive matrices, Vis Comput 40, 2457–2470 (2024).

[16]. Yun S, Han D, Oh S J, et al. Cutmix: Regularization strategy to train strong classifiers with localizable features, IEEE/CVF international conference on computer vision, pp:6023-6032, Seoul, Korea(2019).

[17]. DeVries T, Taylor G W. Improved regularization of convolutional neural networks with cutout. arXiv: 1708.04552(2017).

[18]. Kingma D P, Welling M. Auto-encoding variational bayes, International Conference on Learning Representations, arXiv.1312.6114(2013).

[19]. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets, International Conference on Neural Information Processing Systems, pp:2672-2680, Montreal Canada(2014).

[20]. Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models, International Conference on Neural Information Processing Systems, pp:6840-6851,Vancouver BC Canada(2020).

[21]. Gatys L A, Ecker A S, Bethge M. A neural algorithm of artistic style, arXiv preprint arXiv:1508.06576 (2015).

[22]. Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks, International conference on machine learning, pp:1126-1135, Sydney NSW Australia(2017).

[23]. Cubuk E D, Zoph B, Mane D, et al. Autoaugment: Learning augmentation policies from data, arXiv preprint arXiv:1805.09501 (2018).

[24]. Ma Dongao, Tang Ping, Zhao Lijun, Zhang Zheng, Review of data augmentation for image in deep learning, Journal of Image and Graphics, 26(03):0487-0502(2021).

[25]. Cubuk E D, Zoph B, Shlens J, et al. Randaugment: Practical automated data augmentation with a reduced search space, IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp: 702-703,Seattle, WA(2020).

[26]. Yiming Qin, Nanxuan Zhao, Jiale Yang, Siyuan Pan, Bin Sheng, UrbanEvolver: Function-Aware Urban Layout Regeneration. Int J Comput Vis, 132, 3408-3427 (2024).

[27]. Chen P, Liu S, Zhao H, et al. Gridmask data augmentation, arXiv preprint arXiv: 2001.04086(2020).

[28]. Zezhou Cheng, Qingxiong Yang, Bin Sheng, Deep Colorization. IEEE international conference on computer vision, pp: 415-423(2015).

[29]. Hore A, Ziou D. Image quality metrics: PSNR vs. SSIM, International Conference on Pattern Recognition, pp: 2366-2369, Istanbul, Turkey(2010).

[30]. Wang Z, Simoncelli E P, Bovik A C. Multiscale structural similarity for image quality

assessment, Asilomar Conference on Signals, Systems & Computers, pp: 1398-1402. Pacific Grove, CA(2003).

[31]. Sheikh H R, Bovik A C, De Veciana G. An information fidelity criterion for image quality assessment using natural scene statistics. IEEE Transactions on image processing, 14(12): 2117-2128(2005).

[32]. Everingham M, Van Gool L, Williams C K I, et al. The pascal visual object classes (voc) challenge. International Journal of Computer Vision, 88: 303-338(2010).

[33]. Chen K, Wang J, Pang J, et al. MMDetection: Open MMLab Detection Toolbox and Benchmark, arXiv preprint arXiv:1906.07155(2019).

[34]. Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6): 1137-1149(2017).

[35]. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C], IEEE conference on computer vision and pattern recognition. pp:770-778, Las Vegas, NV(2016).

[36]. Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection, IEEE conference on computer vision and pattern recognition. pp: 2117-2125, Honolulu, HI(2017) .

[37]. Wang C Y, Bochkovskiy A, Liao H Y M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp:7464-7475,Vancouver, BC (2023) .

[38]. Cai Z, Vasconcelos N. Cascade r-cnn: Delving into high quality object detection, Proceedings of the IEEE conference on computer vision and pattern recognition. pp:6154-6162, Salt Lake City (2018) .

[39]. Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection, IEEE international conference on computer vision. pp:2980-2988, Venice, Italy(2017).

[40]. Chen, S., Sun, P., Song, Y., & Luo, P. DiffusionDet: Diffusion model for object detection. arXiv preprint arXiv: 2211.09788 (2022).

[41]. Tan M, Pang R, Le Q V. EfficientDet: Scalable and Efficient Object Detection, IEEE/CVF Conference on Computer Vision and Pattern Recognition .pp: 10781-10790. IEEE(2020).

[42]. Tan M, Le Q V. EfficientNetV2: Smaller Models and Faster Training. International conference on machine learning. PMLR, pp: 10096-10106 (2021).

[43]. Dosovitskiy A, Beyer L, Kolesnikov A, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, arXiv preprint arXiv:2010.11929(2020).