Convergence Acceleration For DiffusionDet: On PROPS Dataset

1st Gongxing Yu
Electrical and Computer Engineering
University of Michigan
Ann Arbor,MI,USA
ethanyu@umich.edu

2nd Liangkun Sun
Electrical and Computer Engineering
University of Michigan
Ann Arbor,MI,USA
liangkun@umich.edu

3rd Yang Lyu
Electrical and Computer Engineering
University of Michigan
Ann Arbor,MI,USA
lyuyang@umich.edu

***Abstract*—*DiffusionDet* is a recent object-detection framework that treats bounding-box prediction as a denoising process: Gaussian noise is added to ground-truth boxes and the network learns to recover the originals. While the authors trained and evaluated DiffusionDet on the large-scale MS-COCO dataset, we investigate its robustness on the much smaller PROPS dataset.In our experiments DiffusionDet converged slowly and achieved sub-optimal accuracy on PROPS, so we added heat-map head after encoder backbone network and introduced an additional heat-map loss to complement the original objective. The modified network converges markedly faster and improves performance by +1.8 AP over the baseline on PROPS.Source code is available at: DiffusionDet-on-PROPS. (Project page: https://deeprob.org)**

I. Introduction

Object detection is a classic computer vision task that aims to locate objects in a digital image. Object detection methods have been evolving with the development of object candidates, for example from fast-RCNN [1],YOLO [2] to learnable object queries such as end-to-end detection with transformers [3] and DETR [4]. Most object detection methods use surrogate regression and classification on empirically designed object candidates, refine the model by minimizing classification and regression errors,*i,e.* Faster-RCNN [1], sliding window based detection [5] etc.

Existing models usually have fix sized learneable queries or bounding boxes, the authors of DiffusionDet [6] proposed a new diffusion-based method that follows the "*noise to box*" paradigm. Similarly to image denoising task, which can generate the image by gradually removing noise from an image via the learned denoising model, DiffusionDet generates the positions (center coordinates) and sizes (widths and heights) of bounding boxes in the image. At the training stage, Gaussian noises are added to ground truth boxes, then use these random boxes as **ROI**(regions of interest) on feature maps extracted by backbone networks(the authors use ResNet [7] and Swin Transformer [8]),then send ROIs to detection decoder, which is a RNN head trained to predict the ground-truth boxes without noise. Thus the model learned how to "*denoise*" random boxes and find ground truth ones. At the inference stage, the model predicts the noise added to bounding boxes at each time stamp all the way to the original ground truth boxes. The original model pipeline is shown in Fig.1

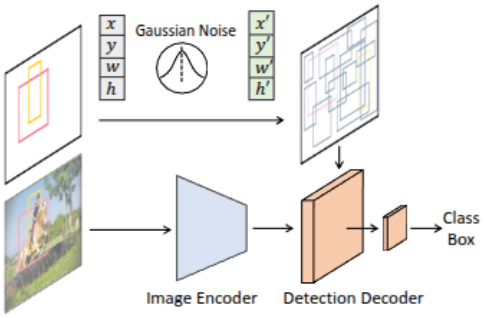

Fig. 1. Original Pipeline of DiffusionDet

However, the original DiffusionDet has some potential drawbacks: First, the authors claim that the model reaches 45.8 AP on COCO dataset(with 200K+ images and over 100 classes), but it cannot prove that DiffusionDet has a good performance on smaller datasets, furthermore, the model may experience some performance fluctuation when detecting small or occluded items in dense scenes due to its complete random bounding boxes. Second, in early stage of training, DiffusionDet generates fixed number boxes as proposals, but many proposals are negative and sparse, which means that the model converges relatively slow in the beginning due to the lack of gradient.

To improve the model in the above 2 ways, we made two contributions as follows:

1) Train and test DiffusionDet on PROPS dataset. This dataset is much smaller(*with only 5K images and 10 classes*) compared to COCO, and contains more challenging detection tasks for DiffusionDet, such as high-frequency occluded objects and small targets. We refine the model by tuning the number of proposals, finally reached 66.67 AP with ResNet-50 backbone, detectron2 [9] learning rate scheduler and 200 proposals.
2) Add a heatmap head after the backbone feature map, designed a weighted heatmap loss and count it toward main loss(box regression and classification). The model obtains a much faster convergence rate(**+10** AP at the first 1000 iterations,**+9** at the first 2000 iterations), higher final AP(**+0.6**).

II. Related Work

A. Object Detection

Object detection is a fundamental task in computer vision, aiming to simultaneously localize and classify objects within an image. Traditional approaches, such as two-stage detectors like Fast R-CNN [1], first generate region proposals and then classify each region. Although effective, these methods often suffer from high computational costs due to their complex pipeline. To address this, single-stage detectors like YOLO [2] were introduced, streamlining the detection process by directly regressing object bounding boxes and class probabilities in a dense prediction manner.

B. Diffusion Models

Diffusion models have recently emerged as a powerful class of generative models, capable of producing high-fidelity samples through a gradual denoising process. Initially proposed for image generation tasks, diffusion models work by reversing a Markovian forward process that adds Gaussian noise to data over multiple time steps [10]. Due to their strong modeling capacity and stable training, diffusion models have achieved state-of-the-art results in image synthesis, super-resolution, and inpainting.

In the context of object detection, diffusion models have been adapted to model the distribution over object bounding boxes and class labels. Instead of treating detection as a direct regression problem, methods like DiffusionDet propose to formulate object detection as a denoising task, where noisy object queries are progressively refined into accurate detections through a learned reverse process [6]. This perspective brings several advantages, including better uncertainty modeling, improved robustness, and the ability to naturally handle variable numbers of objects without the need for predefined anchors or proposals. The integration of diffusion processes into detection frameworks represents a promising direction, bridging generative modeling and structured prediction tasks.

C. DiffusionDet

In the realm of object detection, DiffusionDet [6] introduces a novel generative approach by modeling the detection task as a denoising diffusion process. During training, the model learns to reverse the process of adding noise to ground-truth bounding boxes, effectively learning to recover object boxes from noisy inputs. At inference, DiffusionDet starts from randomly generated bounding boxes and iteratively refines them to accurately detect objects. This method offers flexibility by accommodating a dynamic number of boxes and supports iterative evaluation. Experimental results demonstrate that DiffusionDet achieves favorable performance compared to previous well-established detectors, highlighting the potential of diffusion models in object detection tasks.

D. Heatmap Head

In object detection, the heatmap head is a pivotal component in anchor-free, keypoint-based detectors such as CenterNet [11]. This module predicts a class-specific heatmap over the spatial dimensions of the input image, where each pixel indicates the likelihood of an object center belonging to a particular class. During training, the model encodes ground truth object centers as 2D Gaussians, allowing it to learn spatial attention toward central regions. At inference, high-confidence peaks on the predicted heatmaps correspond to object centers, from which bounding boxes are recovered using additional regression heads.

This design significantly simplifies the detection pipeline by eliminating anchor box generation and non-maximum suppression across predefined regions. Instead, it adopts a keypoint-based representation that is more flexible and interpretable. The heatmap head is typically paired with two additional heads: a dimension head that regresses the width and height of the object, and an offset head that refines the center prediction to counteract resolution loss due to downsampling.

The heatmap-based formulation has demonstrated strong performance in dense detection tasks and has been particularly effective for detecting small or overlapping objects, such as in pedestrian or crowd datasets. Moreover, its lightweight architecture and fast inference capabilities make it well-suited for real-time applications. Subsequent works have further extended this concept to 3D detection, pose estimation, and tracking, confirming the generality and robustness of the heatmap head design.

III. Algorithmic Extension

In order to accelerate convergence rate and improve the model's performance on small or occluded objects, we proposed the heatmap layer. The detailed structure of our optimized DiffusionDet is shown as Fig.2. Heatmap head is composed of 1 2D convolution layer, 1 Relu activation layer and the final convolution layer. The first convolution layer takes high dimension feature map extracted by backbone network and returns 64 dimension filters. The last convolution layer takes activated 64 dimension map to 10 dimension heatmap *w.r.t* class numbers. The predicted heatmap will be interpolated to $\mathbf{C} \times \mathbf{H} \times \mathbf{W}$, where H and W stands for height and width of original image, this is for convenience of training process. We use synthesized heatmap for training.Here is how we do it:

First, consider a ground truth box,its top left corner is at (x_1, y_1), bottom right (x_2, y_2), $(x_1, y_1, x_2, y_2) \in [0, 1]$, class labels $\mathcal{C} \in [0, 1, ..., 9]$, size of heatmap is $H \times W$, we can deride the center coordinates:

$$(x_c, y_c) = (\frac{x_1 + x_2}{2} \cdot H, \frac{y_1 + y_2}{2} \cdot W),$$

For each class label $\mathbf{c}$, we define a 2D Gaussian kernel function to define the "*possibility*" that a point(x, y) is at the center of a ground truth box:

$$G(x, y) = \exp\left(-\frac{(x - x_c)^2 + (y - y_c)^2}{2r^2}\right), \qquad (1)$$

where

$$r = \max(\text{min\_radius},\ \min(w, h) \cdot \rho).$$

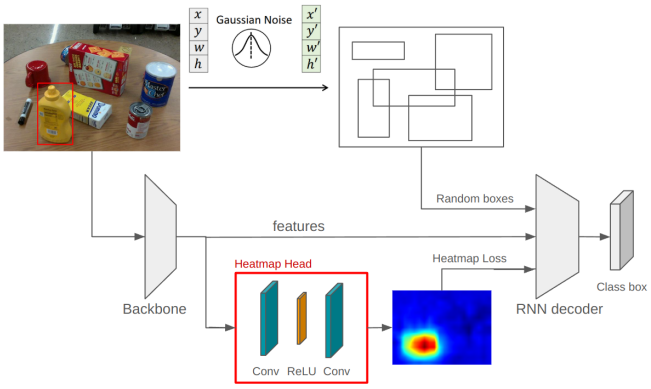

Fig. 2. Our model structure

min_radius is a preset minimum radius that sets a threshold of the size of the "*plateau*" on pixel scale, ρ is a scaling factor. We then normalize the heatmap by dividing the largest value. After experiments, we find that min_radius = 4 and $\rho = 0.5$ will synthesize an accurate heatmap.

To train the heatmap head, we use MSE loss, which is the element-wise mean squared error. As shown in Fig.2, the predicted heatmap of class "Mustard_bottle" is at the bottom right corner, suppose $f(x_p, y_p, C)$ is the value on (x_p, y_p) for class C, z_s is the value at the same position on synthesized heatmap, the MSE loss should be:

$$\mathcal{L}_{min} = \frac{1}{2}||f(x_p, y_p, C) - z_s||^2 \tag{2}$$

Then we multiply the heatmap loss with weight, count it toward total loss generated by RNN decoder. This process is analogous to "space attention" , this gives the model a special hint that "this position is probably occupied by class 0 object", we can calculate gradient from every pixel and each target bounding box is supervised, furthermore, this module is independent from the overall DiffusionDet pipeline, makes it robust and stable.

The heatmap head also improves the model's performance on detecting occluded and small objects. Recall that heatmap layer will predict positions for every object on a image, while in traditional DiffusionDet , small or occluded objects are always neglected by larger targets due to larger bounding boxes.

IV. Experiments and Results

A. Experimental Setup

The experiments are carried out on the PROPS dataset, consisting of 5,000 images in 10 object categories. The model was trained and test on a single NVIDIA A40 GPU. Initially, we implemented DiffusionDet using a Swin Transformer backbone with 30 bounding-box proposals, but observed a notably slow convergence rate. Subsequently, we switched to a ResNet-50 backbone and increased the number of proposals, resulting in improved performance. To further accelerate convergence and boost accuracy, we introduced a heatmap layer with minimum radius of 4 and a scaling factor of 0.5.

The model is trained for 15,000 iterations, with performance evaluated at intervals of 1,000 iterations. Detection accuracy was assessed using mean Average Precision (mAP) over Intersection over Union (IoU) thresholds ranging from 0.50 to 0.95.

B. Results

Fig. 3 compares the Average Precision (AP) over the course of training. The baseline using the Swin Transformer backbone with 30 proposals demonstrated a low initial convergence rate. Upon transitioning to the ResNet-50 backbone with an increased number of proposals, the AP improved considerably. Incorporating the heatmap head led to a substantial acceleration in convergence, especially evident in the early training stages, where AP sharply rose from approximately 38% to 58% within the first 1,000 iterations. After 15,000 iterations, the heatmap-enhanced model achieved a final AP around 67%, surpassing the baseline approaches by approximately 10%.

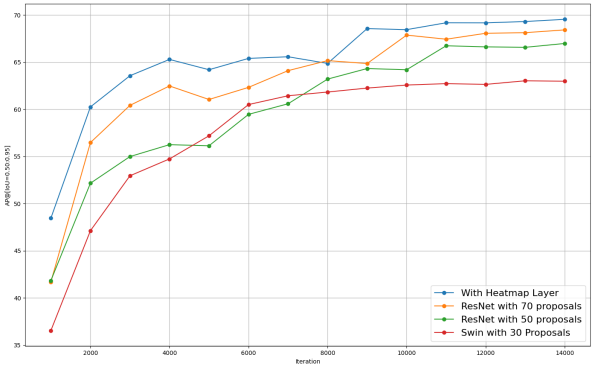

Fig. 3. AP with iteration

Fig. 4 provides qualitative results on sample images from the PROPS dataset, highlighting the heatmap-enhanced model's capability to accurately localize objects with high confidence, including challenging small and partially occluded targets. The optimized DiffusionDet model enhanced by the heatmap layer demonstrates precise localization and exceptionally high detection confidence across various object categories. Notably, the model achieved over 95% confidence scores for all detected instances, illustrating the robustness and accuracy of our proposed heatmap guidance. Table I presents the Average Precision (AP) for each category in the PROPS dataset after adding the heatmap layer. Most object categories achieved AP values above 75. However, detection performance for smaller objects, such as *large_marker* and *tuna_fish_can*, still remains low.

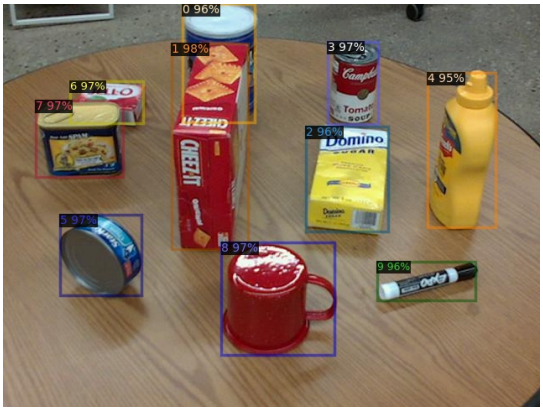

Fig. 4. Result with heatmap on PROPS dataset

TABLE I
PER-CATEGORY AVERAGE PRECISION (AP) ON PROPS DATASET

Category	AP (Without Heatmap)	AP (Heatmap)
master_chef_can	74.873	80.74
cracker_box	72.318	77.84
sugar_box	75.262	79.01
tomato_soup_can	72.893	76.05
mustard_bottle	72.541	79.95
tuna_fish_can	44.210	58.62
gelatin_box	55.481	60.85
potted_meat_can	65.272	70.77
mug	70.459	75.43
large_marker	25.211	34.40
Average AP	62.852	69.368

V. CONCLUSIONS

We evaluated **DiffusionDet** on the PROPS dataset. After tuning the number of box proposals and testing several back-bone networks, the detector reached a respectable mean Average Precision (mAP) of **67

Diffusion-based detectors usually suffer from slow inference and high computational cost. Our proposed **heat-map head** can accelerate inference by reducing the required sampling steps: given a heat-map, the model only denoises boundingboxes in regions with large values—that is, regions that are more likely to contain an object. Because of time constraints we were not able to integrate this module into the inference stage. Nevertheless, the heat-map can be regarded as an explicit prompt for diffusion models, and the same idea might generalise to other tasks such as image generation.

VI. CONTRIBUTION

Gongxing Yu add the heatmap head to to the model,run the experiment and write *introduction* and *algorithmic extensions* sections of this report. Liangkun Sun run the experiment,draw the plot for the results and write the *experiments and results* section of this report. Yang Lyu run the experiment,draw the poster,make the website and finish the rest.

REFERENCES

[1] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[4] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[6] S. Chen, P. Sun, Y. Song, and P. Luo, "Diffusiondet: Diffusion model for object detection," *arXiv preprint arXiv:2211.09788*, 2022.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[9] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[10] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.

[11] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6568–6577.