BiSlim-6D:A 6D pose estimation network for efficient feature decoupling and fusion

Xiaotong Gu[1#], Yongshuai He[2#] and Shenghai Wang[1(✉)]

[1] Marine Engineering College, Dalian Maritime University, Dalian, China
[2] College of Marine Electrical Engineering, Dalian Maritime University, Dalian, China
shenghai_wang@dlmu.edu.cn

Abstract. Pose estimation is an essential component of robotic interaction tasks and remains an active research direction in the field of computer vision. Therefore, developing a 6D pose estimation algorithm that achieves both high speed and high accuracy is crucial. In recent years, leveraging real-time and accurate methods from the YOLO series has become a trend in 6D pose estimation tasks. In this work, we propose a novel 6D pose estimation network called BiSlim-6D, which uses CSPDarknet as the backbone and integrates Slim-neck with BiFPN as the feature fusion network BiSlim-neck. Additionally, we redesign the loss function based on the loss function in YOLO6D, naming it PLoss. Combining these two points, we propose our pose estimation network BiSlim-6D. Finally, through ablation and comparative experiments, we demonstrate that BiSlim-6D exhibits strong overall performance among current 6D pose estimation networks and validate the positive contribution of our restructured components to network performance. The proposed method achieves 98.78% on the 2D reprojection metric and 81.51% on the ADD(-S) metric, with a network inference speed of up to 43.26 FPS. These results fully demonstrate the practical potential of the proposed method in future relevant tasks.

Keywords: Deep Learning, Computer Vision, 6d Pose Estimation, Feature Fusion.

1 Introduction

The task of 6D pose estimation has long been regarded as a core component of technologies such as robotic environment perception, augmented reality, and industrial measurement [1]. For a long time, this task has relied on traditional machine vision methods, such as [2] and [3]. With the emergence of deep learning technology, current research indicates that deep learning-based approaches for 6D pose estimation far outperform traditional methods. Currently, deep learning-based pose estimation methods can be categorized into two main types: those based on depth information and those based on

#These authors contributed to the work equally and should be regarded as co-first authors.

This work was financially supported by the National Natural Science Foundation of China (Project No. 52101396), the National Key Research and Development Program of China (Project No. 2022YFB4300802), and the China Fundamental Research Funds for the Central Universities (Project No.3132023630).

RGB information. While there are some methods based on depth information, such as those relying on RGB-D cameras such as DenseFusion[4], their limitations lie in high energy consumption and suitability only for specific indoor scenarios. In contrast, methods based on RGB cameras such as Bb8[5] have greater advantages in terms of energy consumption and applicability.

Meanwhile, although current deep learning-based methods achieve considerable accuracy in this task, deployment in real-world scenarios still presents many challenges compared to traditional machine vision methods, due to the computational cost and lower real-time performance. One important reason for this is that most current 6D pose estimation methods involve two-shot or more approaches. Tekin et al. proposed a real-time single-shot 6D pose estimation method (YOLO6D[6]) based on YOLOv2[7]. However, due to the rough design of network structure and loss function, the accuracy of the algorithm is relatively low.

Addressing the above issues, this study aims to develop a real-time 6D pose detection algorithm based on RGB images to achieve efficient and accurate detection of object poses. Our method draws inspiration from the rough process of YOLO6D, establishing correspondence between known object 3D models and 2D pixel positions of keypoints, then using a neural network to regress perspective keypoints, and finally utilizing the PnP algorithm to calculate the six pose parameters. We employ the Linemod dataset as a benchmark and propose a real-time, accurate 6D pose estimation network called Slim-6D by reconstructing Slim-neck[8]. Compared with the existing methods, our algorithm not only improves the detection accuracy, but also reduces the energy consumption and calculation cost. In the experimental verification, we verify the effectiveness and real-time performance of the algorithm in chaotic scenarios, demonstrating the potential of deploying the method on edge computing devices for industrial applications.

Our contributions are as follows:

- Reconstruction of BiFPN[9] Structure with Slim-neck Paradigm: We reconstructed the BiFPN structure using the Slim-neck paradigm, tailoring it to suit the requirements of our real-time 6D pose estimation network, BiSlim-6D, resulting in a novel BiSlim-neck structure. By combining the CSPDarknet53 feature extraction network with this reconstructed BiFPN structure, we developed a robust and efficient network architecture called BiSlim-6D. This innovative approach enhances the network's ability to extract and fuse features across different scales, contributing to the rapid and accurate estimation of object poses.
- Development of Real-time 6D Pose Estimation Network - BiSlim-6D: In order to accelerate the inference speed of the network, we use the GSConv module in Slim-neck to reconstruct the network, which effectively improves the real-time performance of the network. Additionally, we redesigned the loss function based on insights from YOLO6D, resulting in Ploss, further optimizing the network's performance on 6D pose estimation tasks. The resulting BiSlim-6D network demonstrates high accuracy, scalability, and robustness, making it suitable for various real-world applications.
- Comparative and Ablation Experiments: We conducted comprehensive comparative experiments on the Linemod dataset[10], evaluating the accuracy and real-time performance of our BiSlim-6D method against other similar approaches. Through these experiments, we demonstrated the effectiveness of our proposed

method. Additionally, we performed ablation experiments to assess the impact of different components of our method on the network performance. Finally, by visualizing the network's inference results, we provided an intuitive understanding of its pose estimation capabilities.

2 Related Works

Now we review existing works on 6D pose estimation. Traditional methods such as [11][12][13] can perform detection tasks in situations with high resolution and relatively rich object textures. However, their robustness in adverse conditions is weak. To address this issue, researchers began to explore the use of Convolutional Neural Networks (CNN) for this task, with PoseCNN[14] being a classic example, which directly regresses pose results using neural networks. Subsequently, there emerged a series of methods that improve algorithm performance by employing complex post-processing techniques, often drawing inspiration from traditional methods. For example, Bb8,Segmentation Driven[15] and YOLO-6Duse CNN to predict the key points of the target, and then return the 6D pose of the target through a series of post-processing processes. Currently, the most mainstream neural network-based 6D pose estimation methods are divided into template-based methods, voting methods, and keypoint-based methods. Template-based methods such as [16] pre-collect images of the target object from various viewpoints and provide poses, transforming the estimation process into an image matching problem, which is costly. Voting methods such as PVNet[17] use the RANSAC[18] method to obtain keypoints based on predictions of keypoints for all points in the image, and then use the PnP algorithm to obtain the final pose. However, due to their dense nature, these methods lack real-time performance. Keypoint-based methods [19] directly regress the projected positions of known prior information of 3D points onto the plane and use the PnP algorithm to calculate the position and pose. This is a sparse method, thus ensuring real-time performance.

Currently, in the task of 6D pose estimation, there are several works employing keypoint-based methods that bear resemblance to the YOLO algorithm structure, such as YOLO6D and MFPN[20]. YOLO6D utilizes CSPdarknet as the backbone network and only designs one output head. Although its work has provided significant inspiration to the field of 6D pose estimation, its accuracy is poor due to its rough network structure and loss function design.

MFPN, proposed by Liu et al., combines CSPdarknet and BiFPN, resembling the structure of YOLOv5, and fine-tunes the BiFPN to develop the MFPN structure. However, its speed and accuracy improvement relative to YOLO6D are not significant.

Therefore, based on these works, we further reconstruct the network structure and loss function to meet the accuracy requirements of current industrial scenarios.

3 Methodology

In this section, we will elaborate on our main contributions: Firstly, we base our approach on the construction method of YOLO6D's loss function. We enhance the network's global scene perception by incorporating Intersection over Union (IOU) terms

into the confidence term of the loss function. Next, we restructure the BiFPN structure using the Slim-neck paradigm, introducing our BiSlim-neck as the neck component of the overall network. We provide reasonable explanations for the performance improvements brought about by this enhancement. Finally, we discuss the improvements we have adopted to accelerate network computations. Combining these innovations, we propose a new 6D pose estimation network named BiSlim-6D.

3.1 PLoss: New Loss Function

Our overall architectural approach is primarily inspired by YOLO6D, utilizing parameterization as follows: Given the prior three-dimensional coordinates of 9 key points relative to the object coordinate system, a neural network designed by us regresses the projected positions of these key points on the view. Subsequently, utilizing the aforementioned information combined with the PnP algorithm, the 6D pose of the object is calculated. These 9 key points include the eight corner points of the Object-Oriented Bounding Box as well as the centroid of the model. This approach is generalizable and similar to MFPN, Bb8, and YOLO6D, as illustrated in Fig. 1.

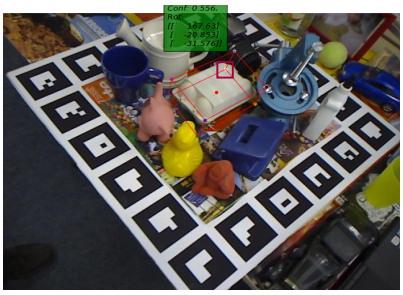

Fig. 1. Hidden spot detection effect diagram

It is worth noting that this approach does not directly regress the true values of image coordinates but rather adopts an indirectly constructed method.

$$f(x) = \left(2\left(\sigma(x)\right) - 0.5\right) + c_{offset} \tag{1}$$

The function is employed, where σ(·) denotes the Sigmoid function with an output range of [0,1], and c_{offset} is an offset used for horizontally shifting the output of the Sigmoid function.

This function scales the output of the Sigmoid function, mapping it from [0,1] to the specified interval, and can be further adjusted through the offset c_{offset} to meet specific requirements.

Additionally, due to the difficulty in calculating the intersection over union of two 3D bounding boxes, we utilize the confidence function proposed by YOLO6D:

$$\mathrm{c}(x) = \begin{cases} e^{\alpha\left(1-\frac{D_T(x)}{d_{th}}\right)}, & if\ D_T(x) < d_{th} \\ 0 & otherwise \end{cases} \tag{2}$$

Here, $D_T(x)$ represents the distance from the model to the detected object at position x, d_{th} is a threshold indicating that the detection result is reliable when the distance is

less than this threshold, and α is a tuning parameter controlling the rate of confidence growth.

When the distance between the detected object and the model is less than the threshold d_{th}, confidence will increase as the distance $D_T(x)$ decreases, with the rate of increase determined by the parameter α. When the distance is greater than or equal to the threshold d_{th} , confidence is zero, indicating that the detection result is unreliable.

It's worth noting that in YOLO6D [reference], it's mentioned that their adopted loss function requires freezing the confidence term's loss in the initial 20 epochs. The fundamental reason behind this is that during the early stages of training, the regression of keypoint positions is inaccurate. However, even in the later stages of training, the reliability of this confidence design method remains suboptimal because it relies solely on precise coordinate information. To address this issue, we incorporate an Intersection over Union (IOU) term into the confidence function:

$$Lconf = IoU\left(max\left(\bigcup_{i=1}^{8} w, \bigcup_{i=1}^{8} h\right)\right) - \left(\frac{\rho^2(b, b^{gt})}{c^2}\right) \tag{3}$$

$$\mathrm{v} = \frac{4}{\pi^2}\left(\arctan\frac{\mathrm{w}^{\mathrm{gt}}}{\mathrm{h}^{\mathrm{gt}}} - \arctan\frac{\mathrm{w}}{\mathrm{h}}\right)^2 \tag{4}$$

$$\alpha = \frac{\mathrm{v}}{(1 - \mathrm{IoU}) + \mathrm{v}} \tag{5}$$

Where the IOU term selects the two farthest points among the nine control points, constructing w and h corresponding to the width and height of these two points in the image coordinate system. Here, α is the weight parameter, and v is used to measure the similarity of aspect ratios. By adding this term to the loss function, the network can estimate the confidence of the cell solely based on the rough positions of the regressed points.

We adopted an improved version of the original IOU loss function. It considers the complete intersection between the target boxes and introduces correction factors to more accurately measure the similarity between them.

Finally, our network utilizes the following loss function during training, which we refer to as PLoss:

$$L = \lambda_{pt} \cdot L_{pt} + \lambda_{conf} \cdot L_{conf} + \lambda_{id} \cdot L_{id} \tag{6}$$

Where L_{pt} is the loss function for position and orientation, orientation. L_{conf} is the loss function for confidence, and L_{id} is the loss function for object identification. λ_{pt} , λ_{conf}, and λ_{id} are the weighting parameters for the position and orientation loss, confidence loss, and object identification loss, respectively.

Due to its similarity to the method of regressing bounding boxes in YOLO, in the inference phase, we employ Non-Maximum Suppression to filter out low-confidence three-dimensional detection boxes from the information regressed in the tensor, ultimately modeling the prediction results.

3.2 BiSlim-neck: More Efficient Real-time Feature Fusion Network

We took inspiration from the concept of the neck part in the YOLO series and incorporate a neck part into our network to effectively represent and process multi-scale features. Efficient representation and processing of multi-scale features in object detection pose a challenge. Early methods typically used pyramid features extracted by backbone networks for target prediction. Feature Pyramid Network (FPN) is a top-down multi-scale feature path, but it can only propagate information from top to bottom and cannot propagate bidirectionally. To address this issue, PANet[21] added additional pathways to FPN for bidirectional propagation, but this increased parameters and computational costs. To improve efficiency and reduce parameters, Google proposed a multi-scale feature fusion method and developed BiFPN, which is more accurate and has lower costs than PANet. As shown in Fig. 2.

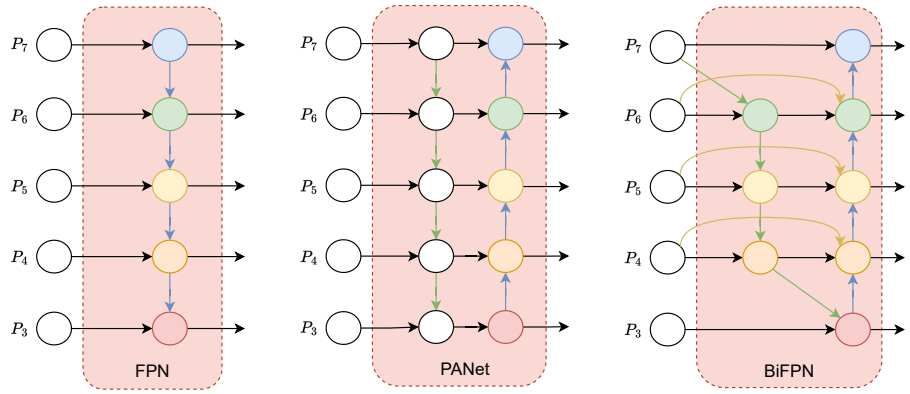

Fig. 2. The network architecture diagrams for Feature Pyramid Network, Path Aggregation Network, and Bidirectional Feature Pyramid Network are shown. We utilize the more efficient and accurate BiFPN.

Due to the significantly greater number of key points our model needs to regress compared to traditional object detection tasks, and the explicit-implicit distinction involved in these feature points (see **Fig. 1**), it undoubtedly implies that our network requires more parameters to achieve a more generalized fit to the data. We adopt the neck construction paradigm Slim-neck proposed by Hulin Li to build our neck section, thereby balancing the increase in parameters and computational complexity resulting from the additional network layers, while improving the detection accuracy. The reconstructed network is referred to as BiSlim-6D, as illustrated in the **Fig. 3**.

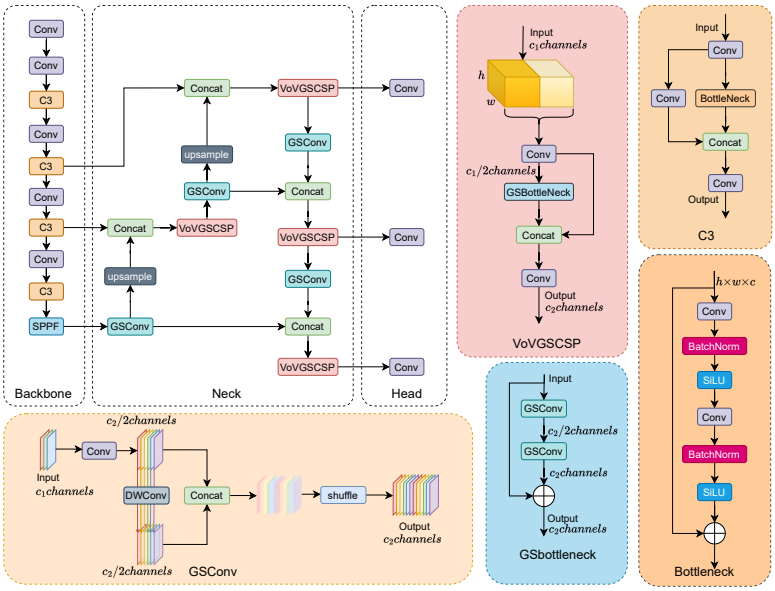

Fig. 3. Improved network structure diagram

Our neural network model consists of three output heads, each producing a tensor of size $(S \times S \times 20)$, where S depends on the downsampling factor of the network. In the last dimension of size 20, corresponding to each cell are the plane coordinates (x_i, y_i) of 9 key points, along with a confidence score and a class label, totaling $2 \times 9 + 1$. This means that each unit in the network output contains predictions of both the probability of the target's presence and its specific location within that region.
We utilize the GSbottleneck and VoVGSCSP modules, as shown in **Fig. 4**..

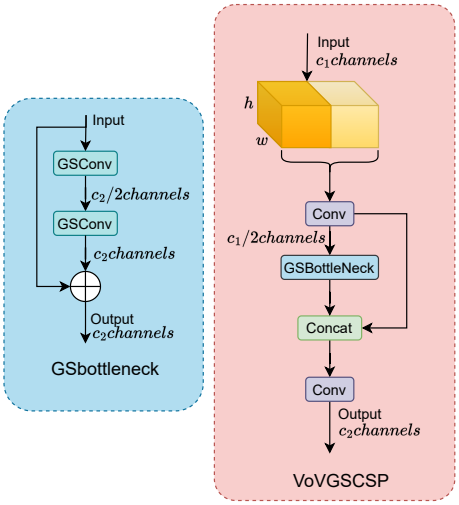

Fig. 4. GSbottleneck module and VoVGSCSP module

The GSbottleneck module is designed using a residual structure, while the VoVGSCSP module is designed using a CSP (Cross-Stage Partial) structure.
Both of these modules consist of the basic unit GSConv, which utilizes shuffling to permeate information generated by SC (channel-dense convolution operations) into various parts of the information generated by DSC (depth-wise separable convolution). Shuffling is a uniform mixing strategy. This approach allows information from SC to be uniformly exchanged across different channels' local feature information and completely mixed into the output of DSC, without cumbersome steps. Fig. 5. illustrates the implementation process of GSConv.

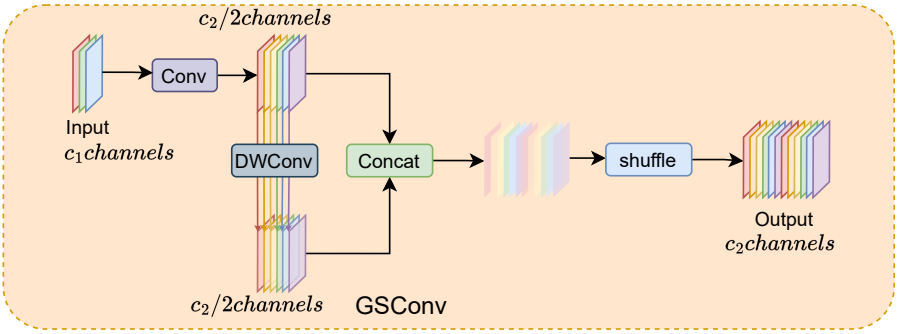

Fig. 5. GSConv network structure

The image feed-forward process in CNN almost always undergoes a similar transformation: spatial information gradually transitions to channels. However, despite this, the information in the channel dimension still retains some degree of the original coupling relationship. As mentioned above (see **Fig. 1**), due to the issue in our task where some key points are implicitly present, their feature representations during network training will adaptively couple with the features of the explicit key points that are easier to regress. In other words, the network doesn't autonomously localize the plane positions of these implicit keypoints. When the explicit feature point localization at the original position is distorted, it inevitably affects the localization of implicit feature points as well. This undoubtedly has a negative impact on the network's generalization. To address this issue, we need to decouple these features in the channel dimension.

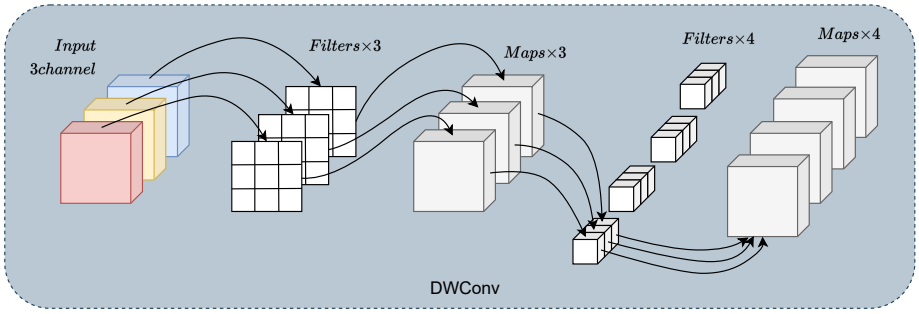

Fig. 6. DWConv network structure

In the GSConv we introduced, the DWConv[22] module decouples the features in the channel dimension, and then the concatenated features undergo shuffling to further decouple and arrange them into a new feature distribution. DSC (Depthwise Separable Convolution) is a convolution operation in convolutional neural networks that consists of two steps: depthwise convolution and pointwise convolution, as shown in Fig. 6.

It can be seen that the key feature of DSC is the separation calculation of channels. This means that DSC first performs convolution operations on each channel of the input, and then performs pointwise convolution on the result of each channel, thereby generating the final output feature map. As mentioned earlier, this channel-separate calculation is beneficial for our task.

As described above, the core module we adopt is the GSConv. This module not only enhances its performance in 6D pose estimation tasks due to its feature decoupling characteristics but also significantly reduces computational and parameter costs, thereby greatly improving the model's inference speed. This is attributed to the DWConv structure within GSConv, which splits the standard convolution into depth-wise convolution and point-wise convolution steps, effectively reducing computational costs.

Consistent with the aforementioned explanation, traditional convolution operations require applying a convolution kernel at every position of all input feature maps. In contrast, DWConv first performs depth-wise convolution on each input feature map and then combines channel features through point-wise convolution. This reduces both computational and parameter costs compared to traditional convolution, as it separates the computation process into two steps. This step-wise computation greatly reduces computational complexity during inference, thus enhancing the model's inference speed.

By reasonably combining the modules for 6D pose estimation tasks mentioned above, BiSlim-6D exhibits good performance and high inference speed in this task, which we will demonstrate in the experiments in the following sections.

4 Experimental design and result analysis

In this section, we will discuss our experimental setup in detail. Firstly, we will introduce the Linemod dataset used and the evaluation metrics employed. Next, we will present our proposed data augmentation strategies. Subsequently, we will showcase and discuss the results of ablation experiments conducted on the Linemod dataset to validate the points proposed in the preceding article and the effectiveness of data augmentation. Finally, we will compare our algorithm with state-of-the-art methods on the Linemod dataset to validate the practical value of our algorithm.

Since we adopt an instance-level pose estimation approach, it is necessary to train separate models for different objects and test their scoring metrics individually. We trained a total of 26 models on different targets, corresponding to different objects in the Linemod dataset. The number of epochs for each model training ranged from 1500 epochs. Through this extensive experimentation, we ensured that our results are highly credible.

4.1 Dataset and evaluation metrics

Linemod dataset. The Linemod dataset is a widely used object recognition dataset in the fields of robotic vision and computer vision research. It has been extensively utilized in research areas such as object recognition, pose estimation, target tracking, etc., and has also become a standard for some competitions, such as the ICRA Robotics Challenge and ECCV Object Recognition Challenge. Therefore, using this dataset to validate our algorithm is credible. The Linemod dataset consists of a number of different objects placed in a variety of cluttered scenes. For each object, there are approximately 1200 images, with each image sized at 640 × 480 pixels. We adopt the same training and testing set division as [11][12][13] to ensure that our experimental results are fair and trustworthy.

Dataset Evaluation Metrics. ADD(Average Distance of Descriptors) is a metric used to evaluate the accuracy of pose estimation algorithms in 3D space. It measures the average distance between descriptors of feature points in two images and is used to assess the quality of feature point matching. Typically, ADD-0.1d is employed, where the pose is considered correct if the average distance of model points is less than 10% of the model diameter. The calculation formula is as follows:

$$ADD = \frac{1}{|M|}\sum_{x\in M}\left\|(Rx+t)-\left(\hat{R}x+\hat{t}\right)\right\|_2 \tag{7}$$

Where:M represents the set of points in the model. $|M|$ denotes the number of points in the model. x_1 and x_2 represent two points in the model, respectively. R and $\hat{R}$ represent the estimated rotation matrix and the ground truth rotation matrix, respectively. t and $\hat{t}$ represent the estimated translation vector and the ground truth translation vector, respectively. $\|\cdot\|_2$ represents the Euclidean distance.

ADD-S (Average Distance of Model Points - Symmetric) is similar to ADD but is specifically designed for symmetric objects. For symmetric objects, traditional ADD metrics may lead to misunderstandings because they calculate the distance from model points to the nearest point. In contrast, the ADD-S metric calculates the average distance from model points to the nearest point, which better reflects the accuracy of symmetric objects. The calculation formula is as follows:

$$ADD-S = \frac{1}{|\mathrm{M}|}\sum_{x_1\in M}\min_{x_2\in M}\left\|(Rx_1+t)-\left(\hat{R}x_2+\hat{t}\right)\right\|_2 \tag{8}$$

Where: M is the set of points on the model. R and t are the estimated rotation matrix and translation vector, respectively. $\hat{R}$ and $\hat{t}$ are the rotation matrix and translation vector of the ground truth pose, respectively. ADD-S evaluates the accuracy of pose estimation by calculating the average minimum Euclidean distance between model points under the estimated pose and model points under the ground truth pose.

2D reprojection error evaluates the accuracy of pose estimation algorithms on the projection plane. It measures the average distance between the projected model points under the estimated pose and the projected model points under the ground truth pose. Specifically, when the distance between projected points is less than a predefined

threshold (usually 5 pixels), we consider the pose estimation to be correct. Let P_e be the estimated camera projection matrix, P_{gt} be the ground truth camera projection matrix, and X be the set of points in the 3D model. The estimated projected points are denoted as $\hat{X} = P_e X$, and the ground truth projected points are denoted as $X_{gt} = P_{gt} X$. The average distance between projected points can be calculated using the following formula:

$$\text{error} = \frac{1}{|\mathrm{X}|} \sum_{\mathrm{x} \in \mathrm{X}} \left\| \hat{x} - x_{gt} \right\| \tag{9}$$

Where:$|X|$ represents the number of points in the model.$\hat{x}$ represents the estimated projected point.x_{gt} represents the ground truth projected point.$|\cdot|$ denotes the Euclidean distance.

4.2 Training

During the training process, we employ a learning rate adjustment strategy that combines warm-up with cosine annealing. In the first three epochs, we use warm-up to gradually increase our learning rate to its initial value. Subsequently, we allow the learning rate to decrease following the cosine function decay until reaching its final value to complete the training.

Additionally, we employ the following data augmentation strategies:

1. Random Erasing: We use a random erasing data augmentation technique to partially occlude some keypoints of the target object. This helps to decouple the network's dependency on specific keypoints during regression and balance the importance of the nine keypoints during regression.

2. Random Scaling and Random Rotation of Images.

The effects of image augmentation are illustrated in **Fig. 7**.

It can be observed that after data augmentation, the images undergo a certain degree of scaling and rotation. Additionally, occlusions are introduced in the images, which are sourced from the PASCAL VOC dataset [46]. Furthermore, our strategy for random erasing differs from traditional random erasing. While still randomly selecting and placing occlusions, our approach ensures that there will always be occlusions covering parts of the target object being trained. As mentioned earlier, this serves as an effective means to decouple explicit keypoints from implicit ones.

Fig. 7. Image effect after data enhancement

4.3 Result analysis

Analysis of ablation results. We conducted ablation experiments on the Linemod dataset, and it is evident that replacing the backbone network leads to varying degrees of improvement in two metrics for different instances. Compared to Darknet, the average 2D reprojection error increased by 3.83% after introducing PLoss. Additionally, before and after the introduction of BiSlim-neck, the network showed a 4.58% improvement in the 2D reprojection error score. This fully demonstrates the effectiveness of our interpretation of the GSConv model proposed in the previous article.

Table 1. Results of 2D reprojection error ablation experiment

Method	Dark-net	CSPdark-net+PLoss	CSPdark-net+BiFPN+PLoss	CSPdarknet+BiSlim-neck+PLoss
Ape	92.10	96.10	97.85	99.14
Benchvise	95.06	97.25	96.72	99.61
Cam	93.14	76.32	96.13	99.51
Can	97.44	86.12	95.89	99.61
Cat	97.41	97.78	92.98	99.30
Driller	79.41	93.28	91.20	98.02
Duck	94.65	99.44	96.71	98.31
Eggbox	90.33	99.87	96.28	99.15
Glue	96.53	99.73	96.22	99.61
Holepuncher	92.86	97.48	93.17	100
Iron	82.94	90.72	94.93	98.47
Lamp	76.87	91.75	95.57	94.43
Phone	86.07	98.81	95.87	99.04
Average	90.37	94.20	95.19	98.78

Table 2. Results of ADD(-S) ablation experiment

Method	Darknet	CSPdarknet+BiFPN	CSPdarknet+BiSlim-neck+PLoss
Ape	21.62	39.89	46.76
Benchvise	81.80	85.70	99.13
Cam	36.57	78.84	82.06
Can	68.8	92.17	91.34
Cat	41.82	53.56	64.47
Driller	63.51	82.98	90.88
Duck	27.23	53.77	35.21
Eggbox	69.58	81.31	99.53
Glue	80.02	76.92	99.81
Holepuncher	42.63	65.87	74.50
Iron	74.97	82.13	93.56
Lamp	71.11	75.94	89.16
Phone	47.74	64.48	93.28
Average	55.95	71.81	81.51

As shown in Table 2, after introducing CSPDarknet and BiFPN, the ADD(-S) score increased to 71.81 compared to the Darknet version, showing an improvement of 15.86.

With the integration of BiSlim-neck proposed in this paper, the ADD(-S) score reached 81.51, demonstrating an enhancement of 25.56 compared to the Darknet version.

Comparative analysis of experimental results.

Pose result score. Our quantitative results on the Linemod dataset are presented in **Table 1** and **Table 2**, while qualitative results are illustrated in Table 3 and **Table 4**. For the ADD(-S) score, we compare our results with those reported in the YOLO6D, PoseCNN, MFPN-6D[23], PVNet[17] and RNNPose[24] papers. It can be observed that our ADD(-S) metric outperforms YOLO6D by an average of 25.56%. Additionally, our method performs only slightly lower than the state-of-the-art method RNNPose 15.59% on this metric. However, as shown in the past of this paper , we will see that BiSlim-6D significantly outperforms RNNPose in terms of real-time performance. Overall, our method demonstrates advantages in scenarios requiring fast computation, making it well-suited for edge computing devices and real-time interactions.

Table 3. ADD(-S) contrast experiment

Method	YOLO6D	PoseCNN	MFPN-6D	RNNPose|pv	BiSliM-6D(ours)
Ape	21.62	25.62	42.65	85.62	46.76
Benchvise	81.80	77.11	87.43	100.0	99.13
Cam	36.57	47.25	81.48	98.43	82.06
Can	68.80	69.98	93.33	99.51	91.34
Cat	41.82	56.09	59.29	96.41	64.47
Driller	63.51	64.92	86.48	99.50	90.88
Duck	27.23	41.74	56.24	89.67	35.21
Eggbox	69.58	98.50	83.40	100.0	99.53
Glue	80.02	94.98	80.95	97.30	99.81
Holepuncher	52.24	42.63	67.74	97.15	74.50
Iron	74.97	70.17	85.87	100.0	93.56
Lamp	71.11	70.73	82.91	100.0	89.16
Phone	47.74	53.07	68.41	98.68	93.28
Average	55.95	63.26	75.08	97.10	81.51

In terms of 2D reprojection error, we compare our results with those reported in the Bb8, MFPN-6D, YOLO6D, and PVNet papers. It is evident that our method, along with MFPN-6D and YOLO6D, exhibits significantly superior metrics. This is attributed to the innovation of our method in network structure and the reconstruction of loss function compared with other methods. Moreover, it can be observed that our method surpasses YOLO6D in 2D reprojection error by 8.00%.

Table 4. 2D reprojection error comparison experiment

Method	Brachmann	Bb8	MFPN	YOLO6D	BiSliM-6D(OURS)
Ape	-	95.3	98.34	92.10	99.14
Benchvise	-	80.0	98.36	95.06	99.61
Cam	-	80.9	97.79	93.14	99.51
Can	-	84.1	97.78	97.44	99.61
Cat	-	97.0	94.75	97.41	99.30
Driller	-	74.1	94.92	79.41	98.02
Duck	-	81.2	97.95	94.65	98.31
Eggbox	-	97.9	98.40	90.33	99.15
Glue	-	89.0	96.28	96.53	99.61
Holepuncher	-	90.5	94.72	92.86	100
Iron	-	78.9	96.57	82.94	98.47
Lamp	-	74.4	97.89	76.87	94.82
Phone	-	77.6	97.85	86.07	93.28
Average	69.5	83.9	97.05	90.37	**98.37**

The above two sets of results sufficiently demonstrate the algorithm's robust capability in both planar and spatial perception. Next, we present more intuitive results of the algorithm. Figure FIG illustrates the detection results of the algorithm on thirteen different object categories in the Linemod dataset. The figure showcases the nine keypoints regressed by the model, the confidence scores of the output results, and the final rotation vectors computed after applying the PnP algorithm.

Fig. 8. Visual rendering of test

FPS performance comparison and evaluation of detection effect. When discussing Frames Per Second (FPS), we compared several different object detection models as shown in **Table 5**. YOLO6D exhibits the highest frame rate, reaching 55.8 FPS, making it the fastest among all methods. PVNet, EfficientPose[25], and EPro-PnPv2[26] achieve frame rates of 30.9 FPS, 27.15 FPS, and 24.9 FPS, respectively, which are

relatively fast but still significantly lower than YOLO6D. RNNPose|pv and Gen6D[27] have lower FPS, with rates of 2.93 FPS and 2.34 FPS, respectively.

Table 5.FPS comparison experiment

Method	YOLO6D	PVNet	Efficient-Pose	EPro-PnPv2	RNNPose|pv	Gen6D	BiSliM-6D(ours)
FPS	55.8	30.9	27.15	24.9	2.93	2.34	43.26

Although the YOLO6D model performs better in terms of FPS, we noticed that its detection performance is not as good as the BiSlim-6D model. This indicates that relying solely on FPS when considering model performance may overlook the importance of detection effectiveness. In contrast, while the BiSlim-6D model is slightly slower than YOLO6D, it exhibits better detection performance, demonstrating higher detection accuracy and robustness. Although RNNPose has better detection accuracy than BiSlim-6D, its lower FPS makes it unsuitable for real-time requirements.

5 Future Work & Limitations

In future research directions, we aim to incorporate the Epropnp method into our proposed algorithm, enhancing its capabilities in 6D pose estimation. Epropnp, as an efficient and robust technique, holds the potential to further improve the accuracy and reliability of our model, especially in challenging scenarios such as occlusions and cluttered environments.

Furthermore, we intend to enhance the generalizability of our algorithm by extending its applicability to a broader range of datasets. Currently, our model relies on instance-specific datasets, limiting its versatility and adaptability to diverse object categories and environmental conditions. By diversifying the training data and incorporating more varied object instances, we aim to bolster the algorithm's ability to generalize across different scenarios and object types.

However, despite the advancements achieved, our algorithm still faces certain limitations. The primary drawback lies in the limited scope of the training dataset, which predominantly consists of instance-specific samples. This restricts the algorithm's generalization capabilities, making it less effective when applied to unseen object categories or environments.

Moreover, while our algorithm demonstrates high accuracy in 2D reprojection metrics, there remains a need to enhance its performance in 3D metrics. Achieving higher precision in 3D pose estimation is crucial for ensuring the robustness and reliability of the algorithm in real-world applications, especially in scenarios where accurate spatial localization is paramount.

Addressing these limitations and pursuing future research directions will be pivotal in advancing the efficacy and applicability of our proposed algorithm in various practical scenarios, ultimately contributing to the broader advancement of 6D pose estimation technology.

6 Conclusion

To improve the accuracy and real-time performance of the 6D pose estimation network, we propose a novel network called BiSlim-6D. Firstly, we redesign the loss function of YOLO6D to reduce its reliance on keypoint regression, thereby enhancing the robustness of confidence calculation. Next, we integrate the BiFPN network with the Slim-neck architecture to construct BiSlim-neck, which enhances the feature extraction capability of the neck network and introduces feature deconstruction characteristics to the network. By combining these two aspects, we obtain the brand new 6D pose estimation network, BiSlim-6D. Additionally, we employ joint data augmentation strategies such as random background replacement, random erasing, and 6D pose transformation to enhance the model's learning of the task's essence. Experimental results demonstrate that compared to similar algorithms, our algorithm achieves a balance between accuracy, computational complexity, and speed, making it suitable for deployment in practical engineering applications.

7 References

1. Du, Guoguang, et al. "Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review." Artificial Intelligence Review 54.3 (2021): 1677-1734.
2. Lowe, D.G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60(2), 91-110.
3. Aldoma, Aitor, et al. "A global hypotheses verification method for 3d object recognition." Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part III 12. Springer Berlin Heidelberg, 2012.
4. Wang C, Xu D, Zhu Y, et al. Densefusion: 6d object pose estimation by iterative dense fusion[C]Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 3343-3352.
5. Rad M, Lepetit V. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth[C]Proceedings of the IEEE international conference on computer vision. 2017: 3828-3836.
6. Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6 DOF pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, (pp.4561-4570).
7. Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
8. Li H, Li J, Wei H, et al. Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles[J]. arXiv preprint arXiv:2206.02424, 2022.
9. Tan M, Pang R, Le Q V. Efficientdet: Scalable and efficient object detection[C]Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 10781-10790.
10. Hinterstoisser S, Lepetit V, Ilic S, et al. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes[C]//Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part I 11. Springer Berlin Heidelberg, 2013: 548-562.

11. Haoruo Zhang and Qixin Cao. Detect in RGB, Optimize in Edge: Accurate 6D Pose Estimation for Texture-less Industrial Parts. IEEE International Conference on Robotics and Automation (ICRA), 2019, (pp.3486-3492).
12. Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stean Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In Asian Conference on Computer Vision (ICCV), 2012, (pp.548-562).
13. Wadim Kehl, Faysto Milletari, Federico Tombari, Slobodan Ilic, and Nassir Navab. Deep Learning of Local RGB-D Patches for 3D Object Detection and 6D Pose Estimation. In Proceedings of the European Conference on Computer Vision (ECCV), 2016, (pp.205-220).
14. Xiang Y, Schmidt T, Narayanan V, et al. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes[J]. arXiv preprint arXiv:1711.00199, 2017.
15. Hu, Yinlin, et al. "Segmentation-driven 6d object pose estimation." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.
16. Hinterstoisser, Stefan, et al. "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes." 2011 international conference on computer vision. IEEE, 2011.
17. Peng, Sida, et al. "Pvnet: Pixel-wise voting network for 6dof pose estimation." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.
18. Fischler, Martin A., and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." Communications of the ACM 24.6 (1981): 381-395.
19. Tekin, Bugra, Sudipta N. Sinha, and Pascal Fua. "Real-time seamless single shot 6d object pose prediction." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
20. Liang, Tingting, et al. "MFPN: A novel mixture feature pyramid network of multiple architectures for object detection." arXiv preprint arXiv:1912.09748 (2019).
21. Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8759-8768.
22. Chollet, François. "Xception: Deep learning with depthwise separable convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
23. Liu, Penglei, et al. "MFPN-6D: Real-time One-stage Pose Estimation of Objects on RGB Images." 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021.
24. Xu, Yan, et al. "Rnnpose: Recurrent 6-dof object pose refinement with robust correspondence field estimation and pose optimization." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
25. Bukschat Y, Vetter M. EfficientPose: An efficient, accurate and scalable end-to-end 6D multi object pose estimation approach[J]. arXiv preprint arXiv:2011.04307, 2020.
26. Chen H, Wang P, Wang F, et al. Epro-pnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 2781-2790.
27. Liu Y, Wen Y, Peng S, et al. Gen6d: Generalizable model-free 6-dof object pose estimation from rgb images[C]//European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022: 298-315.