electronics

MDPI

Article

FusionNetV2: Explicit Enhancement of Edge Features for 6D Object Pose Estimation

Yuning Ye [1] and Hanhoon Park [1,2,*]

[1] Department of Artificial Intelligence Convergence, Graduate School, Pukyong National University, 45 Yongso-ro, Nam-gu, Busan 48513, Republic of Korea; yeyuning12@gmail.com
[2] Division of Electronics and Communications Engineering, Pukyong National University, 45 Yongso-ro, Nam-gu, Busan 48513, Republic of Korea
* Correspondence: hanhoon.park@pknu.ac.kr; Tel.: +82-51-629-6225

Abstract: FusionNet is a hybrid model that incorporates convolutional neural networks and Transformers, achieving state-of-the-art performance in 6D object pose estimation while significantly reducing the number of model parameters. Our study reveals that FusionNet has local and global attention mechanisms for enhancing deep features in two paths and the attention mechanisms play a role in implicitly enhancing features around object edges. We found that enhancing the features around object edges was the main reason for the performance improvement in 6D object pose estimation. Therefore, in this study, we attempt to enhance the features around object edges explicitly and intuitively. To this end, an edge boosting block (EBB) is introduced that replaces the attention blocks responsible for local attention in FusionNet. EBB is lightweight and can be directly applied to FusionNet with minimal modifications. EBB significantly improved the performance of FusionNet in 6D object pose estimation in experiments on the LINEMOD dataset.

Keywords: object pose estimation; convolutional neural network; Transformer; hybrid model; edge boosting

Citation: Ye, Y.; Park, H. FusionNetV2: Explicit Enhancement of Edge Features for 6D Object Pose Estimation. *Electronics* **2024**, *13*, 3736. https://doi.org/10.3390/electronics13183736

Academic Editors: Taehyeon Kim and KyungTaek Lee

Received: 22 August 2024
Revised: 9 September 2024
Accepted: 19 September 2024
Published: 20 September 2024

1. Introduction

In computer vision applications such as robotics and augmented reality, accurately determining the six-degree-of-freedom (6D) pose of objects in relation to the camera (encompassing 3D rotation and translation) stands as a fundamental task. Deep learning has revolutionized this, similar to other vision tasks. Popular approaches involve feature point matching and perspective-n-point (PnP) algorithms, where convolutional neural networks (CNNs) play a central role in feature extraction and pose prediction [1]. CNNs form the cornerstone of deep learning models tailored for computer vision tasks. Their prowess lies in effectively capturing local spatial features. As deep learning continues to thrive in the field of computer vision, conventional components such as non-maximal suppression and region of interest cropping have been substituted with superior alternatives, facilitating the development of end-to-end differentiable pipelines [2]. Backbone networks based on CNNs have emerged as a prevalent and dominant approach across various vision tasks including 6D object pose estimation [3]. However, it is worth noting that the convolutional operation processes one local neighborhood at a time, which makes it less capable of capturing long-range dependencies between features, posing a challenge to understanding the global context of input images. Consequently, numerous studies have explored methods to address and alleviate this issue.

Capturing long-range dependencies is a key consideration in deep neural networks aimed at a holistic understanding of visual scenes [4,5]. In CNN-based deep learning approaches, these dependencies are modeled indirectly by leveraging large receptive fields formed through deep stacks of convolutional operations. However, the repetitive

application of convolutional operations has still proven to have a limited grasp of the global features required in various vision tasks [6,7].

Transformer architecture, originally devised for self-attention in natural language processing (NLP), has recently made significant strides in the field of computer vision, notably exemplified by Vision Transformer (ViT). In ViT, input images are segmented into non-overlapping patches, treated as markers similar to tokens in NLP. These patches, accompanied by special positional encoding for coarse spatial information, undergo processing through repeated standard Transformer layers, effectively capturing global features for image classification [8–10]. A hierarchical ViT has since been proposed, capturing global and local range dependencies and inter-frame dependencies to achieve better performance [11]. However, the performance of ViT-based models is still lower than that of CNN-based models of similar size when trained on small amounts of data [12].

To bring together the benefits of CNNs and Transformers, FusionNet [13] integrates the Transformer architecture into the convolutional architecture. For a given input image, FusionNet extracts informative features through the four-stage CNN backbone and utilizes the Transformer to enhance the features by considering long-range dependencies. Additionally, FusionNet introduces an attention block (AtB) to improve learning on the local context of the CNN backbone. FusionNet fuses the features from both architectures. Combined with the EPro-PnP head [14], FusionNet has shown state-of-the-art performance in 6D object pose estimation.

FusionNet enhances features through the attention mechanisms in two paths (the AtBs in the CNN backbone and the Transformer), improving the ability to learn the local and global context of input images. The attention mechanisms weigh features of different importance and suppress features that are not conducive for 6D object pose estimation. As shown in Figure 1c, the resulting feature maps are highlighted in object edges, indicating a high correlation between object pose estimation and features in the edge regions. This is more apparent in the results of AtBs. Based on these findings, we propose an approach that improves the ability of FusionNet to learn edge features to improve its performance in 6D object pose estimation. In this study, we attempt to enhance features in the edge regions explicitly and intuitively. Thus, we introduce a simple yet effective attention block, Edge Boosting Block (EBB), replace the AtBs in FusionNet with EBBs.

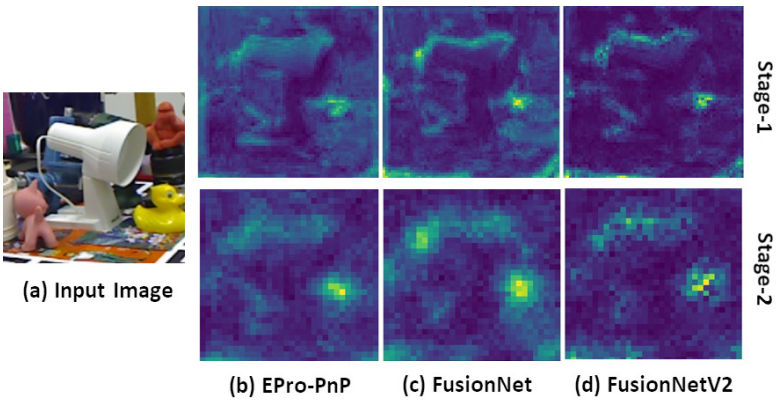

Figure 1. Visualization of feature maps output from Stage 1 and Stage 2 of the CNN backbone of EPro-PnP, FusionNet, and FusionNetV2 for the same input. The feature maps of FusionNet are highlighted around object edges. FusionNetV2 further boosts features around object edges while strongly suppressing the rest.

The primary contributions of this study, which focuses on developing an improved deep learning model for 6D object pose estimation, are as follows:

- We propose FusionNetV2 with the improved ability of FusionNet to learn edge features closely related to 6D object pose estimation.

- We propose an attention block called EBB, which is simple to implement and specialized for enhancing edge features.
- The performance of FusionNetV2 is validated on a benchmark dataset in various aspects. The experiments show that FusionNetV2 outperforms FusionNet in 6D object pose estimation.

The remainder of this paper is organized as follows. In Section 2, the problem that we want to solve in this paper is defined. In Section 3, related works are reviewed. In Section 4, our baseline model, FusionNet, is briefly described. Then, our FusionNetV2 is elaborated in Section 5. In Section 6, the performance of FusionNetV2 is validated on benchmark datasets. Finally, the conclusion and future studies are presented in Section 7.

2. Problem Definition

The 6D object pose estimation refers to the task of determining the 6D pose (representing the position and orientation of an object) of an object in a scene given an input image. Following the framework of EPro-PnP [14], we assume that the input crop image containing the target object is pre-acquired by an object detector. Then, our goal is to predict the 3D object coordinates X_i and weights w_i at the object pixels $x_i (i = 1, \ldots, N)$ using a deep neural network, from which a weighted PnP algorithm is used to estimate the object pose relative to the camera. That is, the rotation matrix R and translation vector t are calculated by minimizing the following:

$$\arg\min_{R,t} \frac{1}{2} \sum_{i=1}^{N} |w_i \{\pi(RX_i + t) - x_i\}|^2, \tag{1}$$

where $\pi(\cdot)$ is the camera projection function.

3. Related Work

The information of depth or point cloud facilitates 6D object pose estimation [15,16]. However, depth or point cloud data are not always available or unaffordable and are usually inaccurate or sparse. In recent years, methods using only RGB images for 6D object pose estimation have been widely studied. In this section, we provide a brief overview of 6D pose estimation methods using only RGB images, categorizing them into CNN-based and Transformer-based methods.

3.1. CNN-Based Method

CNN-based approaches to 6D object pose estimation fall into two main categories: indirect and direct methods. Direct methods directly derive object poses from input images, whereas indirect methods estimate robust intermediate representations, subsequently deducing object poses from these representations. Keypoints serve as a highly popular indirect representation, and some previous methods based on keypoints have yielded impressive results [17–21]. Pavlakos et al. [19] proposed an approach that combines semantic keypoints predicted by a stacked hourglass CNN with a deformable shape model. This method involves the convolutional network learning the optimal representation from available training image data, without taking texture into consideration. Oberweger et al. [22] proposed a technique that anticipates the 2D projections of 3D points associated with the target object. They subsequently compute the 6D pose based on these correspondences using a geometric approach. To address occlusion challenges, they independently predict heatmaps from multiple small patches and aggregate the results for robust predictions. Another representation method similar to keypoints is dense prediction. In this approach, the entire 3D model is projected onto 2D images to establish 2D-3D correspondences. The final pose is then calculated using these correspondences. Haugaard et al. [23] proposed a method that learns a dense, continuous 2D-3D correspondence distribution on the object's surface without requiring prior knowledge of visual ambiguities such as symmetry. The 6D pose is then computed using the PnP algorithm based on the correspondence distribution.

Chen et al. [14] proposed an end-to-end method for 6D object pose estimation, treating the 2D-3D coordinates and corresponding weights as intermediate variables. These variables are learned by minimizing the Kullback–Leibler divergence between the predicted and target pose distribution.

In recent studies, Hai et al. [24] introduced an unsupervised method for 6D object pose estimation. They employed a teacher–student architecture and, during training, generated pixel-level flow supervision signals by leveraging the geometry-guided flow consistency between images from different views. Remarkably, their approach achieved performance comparable to most supervised methods even in an unsupervised scenario. Yang et al. [25] introduced a two-stage 6D object pose estimation method specifically designed for textureless objects. The method predicts both the direction and distance from all pixels within the object's edge representation to specific object keypoints. Through establishing a sparse 2D-3D correspondence based on voting, the method utilizes the PnP algorithm to accurately determine the object's pose. Li et al. [26] proposed a weakly supervised reconstruction-based approach named NeRF-Pose, requiring only 2D bounding boxes and relative camera poses during training. Following the idea of reconstructing first and then regressing, their method initially reconstructs the object from multiple views in the form of an implicit neural representation. Subsequently, a pose regression network is trained to predict pixel-wise 2D-3D correspondences between the image and the reconstructed 3D model. In contrast to previous studies, Wu et al. [27] proposed a method that employs training a graph network to select a dispersed set of keypoints with similar distribution votes, aiming to improve accuracy and efficiency. This deviates from the heuristic keypoint position selection common in previous methods, leading to high-performance improvements.

3.2. Transformer-Based Method

The successful integration of Transformer into vision tasks by ViT has sparked various attempts to apply Transformer in the field of computer vision. Recent studies indicate that Transformer also exhibits competitive performance in 6D object pose estimation. PoET [28] utilizes a Transformer as the backbone. RGB images are inputted to an object detector to generate feature maps and predict objects' bounding boxes. Subsequently, the feature maps are fed into the Transformer, with the detected bounding boxes serving as additional information. The output is processed by separate translation and rotation heads. Similarly, YOLOPose [29], a Transformer-based keypoint regression model for 6D object pose estimation, incorporates bounding boxes as one of the information for pose estimation. Additionally, they jointly estimate labels for all objects in the input image, along with translation parameters and pixel coordinates of 3D keypoints, as supplementary information for precise pose estimation. Trans6D [30] is a Transformer-based framework designed to predict dense 2D-3D correspondence maps from an RGB input image. An additional module called Trans6D+, responsible for pose refinement, is also introduced. This module learns the transformation between the predicted pose and the ground-truth pose and contributes to further improving the performance of Trans6D. CRT-6D [31], a cascade of Transformers for 6D object pose estimation, iteratively refines initial pose estimates by applying self-attention to a set of sparse object keypoint features. FoundationPose [32] is a comprehensive model serving as a foundation for both 6D object pose estimation and tracking, accommodating both model-based and model-free configurations. The use of Transformer-based network architectures and a contrastive learning formulation resulted in strong generalization when trained solely on synthetic data, which allows for immediate application at test time to a new object without the need for fine-tuning, provided that its CAD model is available or a small number of reference images are captured.

3.3. Hybrid Method

The exploration of merging CNN and Transformer architectures to solve low-level or high-level vision problems has been investigated [33–36]. However, to the best of our

knowledge, FusionNet is the first and only attempt to merge the two architectures in 6D object pose estimation, except by simply using CNNs at a pre-processing step for extracting shallow features used as the input in Transformer-based models [28–31]. FusionNet will be briefly reviewed in the next section.

4. Revisiting FusionNet

The key idea of FusionNet [13] is to simultaneously leverage the strengths of CNN and Transformer, integrating them into a hybrid model. FusionNet extracts informative features using multi-stage efficient CNN blocks while concurrently incorporating long-range dependency between features obtained by Transformers. To enhance features and filter out unnecessary information, FusionNet also employs a simplified attention module [7] in the CNN structure. These allow the model to estimate a more accurate 6D object pose while maintaining its lightweight structure. The overall structure is illustrated in Figure 2. It consists of four stages, each containing several CNN blocks, taking RGB images of size 256×256 as input, and generating feature maps at different scales. The output of Stage 2 is fed into a Transformer block, global dependency encoder (GDE), and concatenated with the output of Stage 3. The output of Stage 4 is then input into the EPro-PnP head [14], which consists of two subheads: one for predicting translation parameters using a regression model, and another for extracting dense 3D coordinate maps and weight maps through convolutional layers. A PnP block, replacing the PnP algorithm, is used to predict rotation parameters from 3D-2D coordinate pairs.

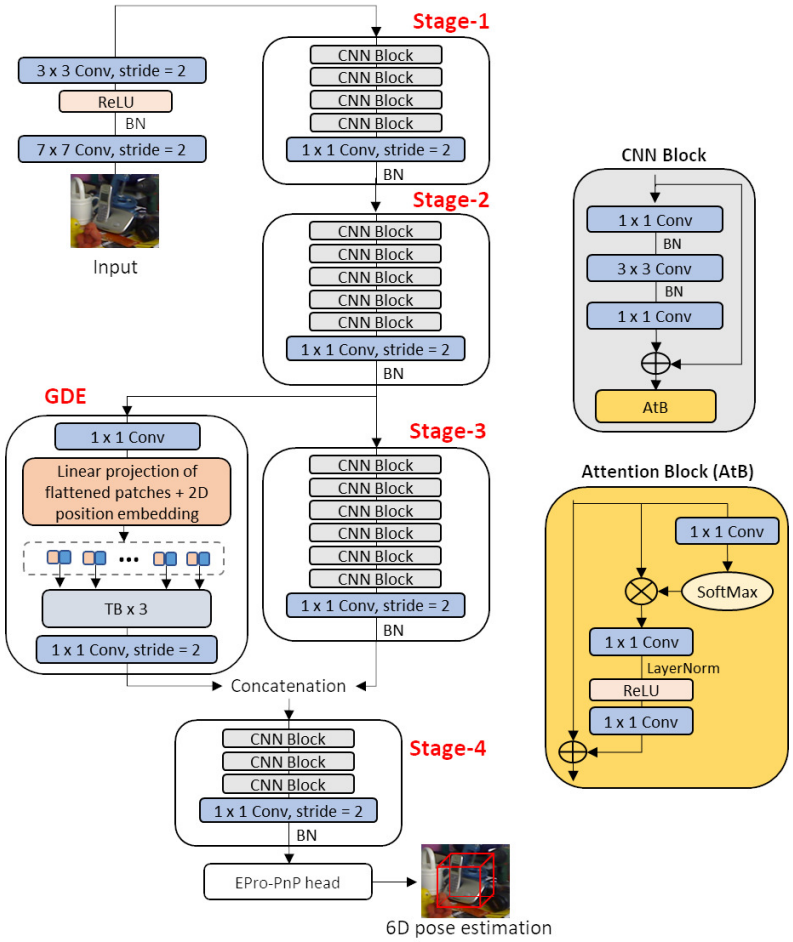

Figure 2. FusionNet's architecture.

In FusionNet, features are enhanced through the attention mechanisms in two paths (the AtBs on the CNN side and the multi-head attention in GDE). The resulting feature maps

are highlighted at object edges (Figure 1), which is a main contributor to the performance improvement in 6D object pose estimation. However, we believe that the implicit approach using the AtBs and GDE has limitations in enhancing the features around object edges. Furthermore, the repeated use of AtB in CNN blocks increases the complexity of the model structure and also reduces flexibility for modifications.

5. FusionNetV2

The purpose of FusionNetV2 is to strengthen the extraction of features around object edges over FusionNet. Therefore, we propose a simple yet effective attention block, named EBB, which explicitly enhances the features around the edges. We eliminate all AtBs used in CNN blocks and add one EBB to the end of Stage 1 and Stage 2, respectively. The overall structure of FusionNetV2 is illustrated in Figure 3.

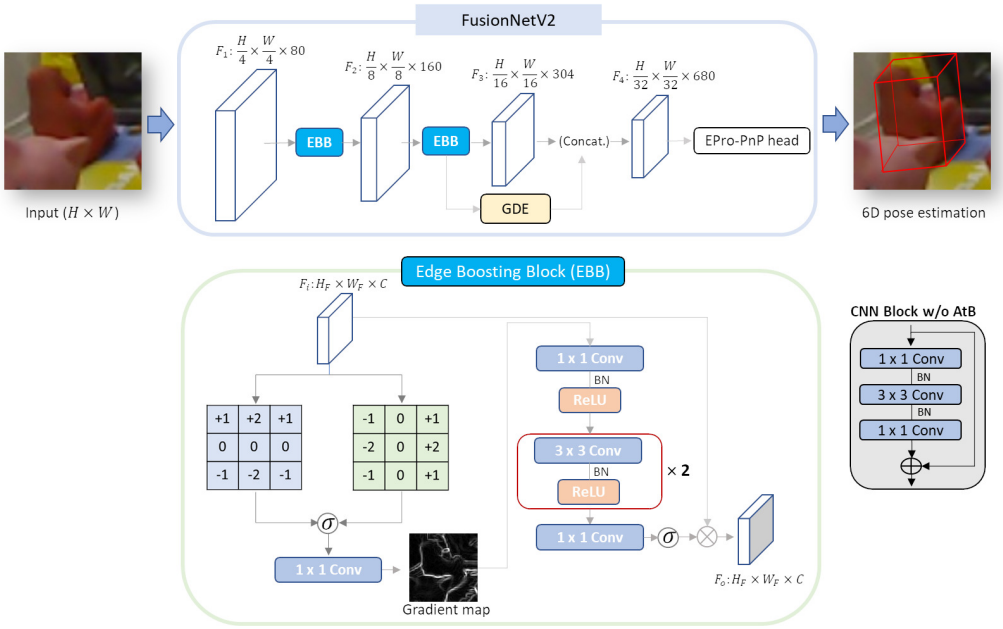

Figure 3. FusionNetV2's architecture. The overall structure of FusionNetV2 is exactly the same as FusionNet, except for removing AtBs in the CNN Blocks and adding two EBBs instead. The resolution of feature maps gradually decreases from high (4 strides) to low (32 strides) over four stages. F_1, F_2, F_3, and F_4 are the output feature maps for each stage; only F_1 and F_2 are enhanced by EBB.

Inspired by [37], EBB utilizes the Sobel operator [38], composed of two 3×3 fixed-parameter convolutions with a stride of 1, to compute gradient maps:

$$C_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad C_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}. \tag{2}$$

The resulting gradient maps (M_x and M_y) are then normalized by a sigmoid function, merged through a convolution layer, and enhanced through three Conv-BN-ReLU layers. The final attention map is fused with the input feature map F_i through element-wise multiplication:

$$F_o = F_i \odot \sigma\big(Convs(\sigma(M_x), \sigma(M_y))\big). \tag{3}$$

Here, $\odot$ represents element-wise multiplication, σ is the sigmoid function, and *Convs* denotes three Conv-BN-ReLU layers. EBB improves the ability to distinguish edge regions from non-edge regions in the feature space and filters out irrelevant details, allowing the network to focus on extracting essential features near the target object's edges.

The reason why EBB is applied to both Stage 1 and Stage 2 is that shallow features inherently contain rich spatial information at higher resolutions, which facilitates the extraction of features closely related to object edges. The object edges in deep features extracted in Stage 3 and Stage 4 are indistinguishable because deep features contain semantic information rather than low-level information (Figure 4). EBB with a small and fixed receptive field is not suitable for extracting the semantic edges. Later in the experimental results, we will show the performance difference of EBB when applied to features with different levels.

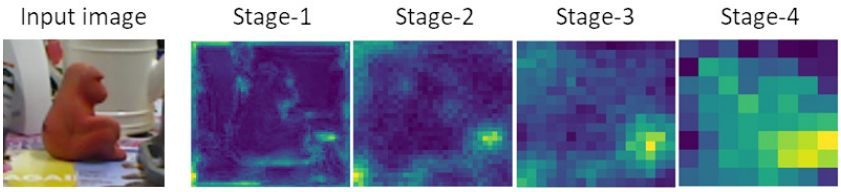

Figure 4. Output features at each stage of FusionNet.

6. Experimental Results and Discussion

6.1. Experimental Setup

FusionNet used EPro-PnP as the baseline model, and FusionNet is the baseline model of FusionNetV2. Therefore, the performance of FusionNetV2 is evaluated by comparing it to EPro-PnP and FusionNet. We believe that the comparison of FusionNetV2 and the other 6D object pose estimation methods is not the main concern of this paper and can be seen from the results in [13]. The main difference between FusionNet and FusionNetV2 is the presence or absence of EBB. First, the performance of FusionNetV2 based on the position and number of EBBs is analyzed in an ablation study. Then, its training stability and generalization ability are analyzed by visualizing the training errors and validation accuracies and compared to that of FusionNet. Next, to show the effectiveness of EBB, another method is attempted to enhance edge features, and its performance for pose estimation is analyzed. Finally, the inference time of FusionNetV2 is analyzed and additional attempts to reduce the inference time are discussed.

We conducted experiments for training and testing the models using the LINEMOD dataset [39], which consists of 13 sequences. Each sequence comprises approximately 1.2 K images with annotations for the 6D pose and 2D bounding box of a single object. Additionally, 3D CAD models are provided for each object. Following the approach in [40], images were divided into training and testing sets, with approximately 200 images per object used for training. The training data were augmented using the synthetic data used in CDPN [41].

We use ADD(-S) and the 2D reprojection error (2DRE) as the evaluation metrics for the final pose. ADD measures whether the average 3D distance between the vertices of the object mesh transformed by the ground-truth pose and the predicted pose falls below a specified fraction of the object's diameter. For example, ADD-0.1d considers the predicted pose accurate when the distance is below 10% of the object diameter. Moreover, 2DRE is the average distance between the 2D projections of the object's 3D mesh vertices under the predicted pose and the ground-truth pose. The prediction is considered accurate if the error is less than 5 pixels. For both metrics, the percentage of images in which the predicted pose is accurate to all test images is measured.

For a fair comparison, the general experimental setup is kept the same as in EPro-PnP [14] and FusionNet. The implementation utilized the open-source codes of EPro-PnP [14] and FusionNet [13], implemented with PyTorch, on a desktop computer (i7 2.9 GHz CPU and 32 GB RAM) equipped with a single RTX 2060 GPU. The source code is accessible at https://github.com/helloyuning/FusionNetV2 (accessed on 21 August 2024). During training, the RMSProp optimizer was employed with parameters $\alpha = 0.99$, $\epsilon = 1 \times 10^{-8}$, $\lambda = 0$, and $\mu = 0$. The learning rate and the number of epochs were set to 1×10^{-4} and 320, respectively.

Unfortunately, due to device constraints, the batch size was reduced from 32 to 16 for training. Under the same conditions, we focus on showing the superiority of FusionNetV2 over EPro-PnP and FusionNet. In our experiments, it was observable that the performance of the fine-tuning strategies used in EPro-PnP and FusionNet is highly dependent on the accuracy of the pre-trained model. To ensure a fair comparison and alleviate the undue impact of pre-training, we opted to train all models from scratch and evaluated their performances comparatively.

6.2. Ablation Study

Compared to FusionNet, FusionNetV2 introduces EBBs to explicitly enhance features on object edges. However, its performance varies depending on the location of the EBB. Therefore, the impact of having an EBB at different stages of FusionNetV2 on model performance is analyzed. In the tables below, "s-x,y" represents that the EBBs are located at the end of Stage x and Stage y. Table 1 shows the resulting ADD scores of having EBBs at different stages. From the results, it can be observed that:

- The accuracy of EPro-PnP and FusionNet was not so high without pre-training. Their ADD-0.1d scores remained at 73.78 and 83.07, respectively, although FusionNet could significantly increase the ADD-0.1d score by employing GDE and AtBs.
- The accuracy of FusionNetV2 was dependent on which stage the EBB is placed. Comparing the results of placing EBB in a single stage only, placing EBB in Stage 2 was the most effective, with an ADD-0.1d score reaching 87.09. Even with EBB in only one stage, the ADD-0.1d score was approximately 4 points higher than FusionNet. However, placing one EBB in Stage 1 or Stage 3 could not achieve ADD scores equivalent to FusionNet.
- Comparing the results of placing EBBs over multiple stages, additionally placing EBB in Stage 3 or Stage 4 did not help improve accuracy but rather may impair accuracy. It indicates that applying EBBs with a small receptive field to deep features associated with semantic information can lead to the loss of important semantic information. Placing EBBs in Stage 1 and Stage 2 (the stages responsible for extracting shallow features) achieved the highest accuracy, with an ADD-0.1d score reaching 90.18. FusionNetV2 with EBBs in Stage 1 and Stage 2 achieved mean ADD scores approximately 18.8 and 9.7 points higher than EPro-PnP and FusionNet, respectively.

Table 1. Ablation study of FusionNetV2. The values within parentheses denote the degree of improvement achieved with each modification.

	ADD(-S)			
	0.02d	**0.05d**	**0.1d**	**Mean**
EPro-PnP	**12.05**	**43.79**	**73.78**	**43.37**
FusionNet	19.32 (+6.78)	55.15 (+11.36)	83.07 (+9.29)	52.51 (+9.14)
FusionNetV2 (s-1)	15.3 (+3.25)	49.18 (+5.39)	77.62 (+3.84)	47.37 (3)
FusionNetV2 (s-2)	22.62 (+10.57)	61.11 (+17.32)	87.09 (+13.31)	57.21(+13.84)
FusionNetV2 (s-3)	17.52 (+5.47)	52.52 (+8.73)	80.03 (+6.25)	50.02 (+6.65)
FusionNetV2 (s-1,2)	**28.54** (+16.49)	**67.79** (+23.9)	**90.18** (+16.4)	**62.17** (+18.8)
FusionNetV2 (s-1,2,3)	18.16 (+6.11)	53.81 (+10.32)	81.68 (+7.9)	51 (+7.63)
FusionNetV2 (s-1,2,3,4)	16.8 (+4.75)	53.59 (+9.8)	81.33 (+7.55)	50.57 (+7.2)

To further analyze the impact of placing EBBs at different stages, the ADD-0.1d scores for each object (Table 2) were compared. For all objects, using EBB only at one stage improved ADD-0.1d scores compared to EPro-PnP. However, using one EBB at Stage 1 or Stage 3 had lower ADD-0.1d scores than FusionNet. It is shown that EBBs must be placed at Stage 2 in order to obtain higher ADD-0.1d scores than FusionNet. When EBBs were placed at Stage 1 and Stage 2, ADD-0.1d scores were the highest except for "Can" and "Eggbox". The variance of ADD-0.1d scores over objects was also the smallest when EBBs were placed

at Stage 1 and Stage 2, and smaller than FusionNet, indicating that the performance was more stable and reliable.

Table 2. Object-wise ADD-0.1d scores of EPro-PnP, FusionNet, and FusionNetV2.

	EPro-PnP	FusionNet	FusionNetV2					
			s-1	s-2	s-3	s-1,2	s-1,2,3	s-1,2,3,4
Ape	53.14	64.29	52.76	60.48	54.67	**76.19**	59.90	62.19
Bench vise	88.17	90.20	88.94	93.50	90.59	**95.93**	90.69	89.72
Camera	65.49	79.02	72.74	85.78	77.06	**90.00**	77.25	77.65
Can	75.10	84.25	80.22	**95.47**	82.97	92.22	84.45	89.57
Cat	58.38	74.65	65.07	80.64	70.56	**85.13**	71.56	78.14
Driller	78.39	86.92	82.85	92.17	87.12	**93.46**	86.32	87.71
Duck	60.28	66.67	58.50	70.99	54.65	**77.37**	64.13	38.69
Egg box	97.56	99.25	99.34	99.34	99.15	99.34	**99.44**	98.78
Glue	80.12	89.67	85.81	94.59	88.80	**95.37**	90.15	93.34
Hole puncher	62.32	78.40	66.79	82.21	71.36	**87.63**	70.50	75.74
Iron	87.54	92.44	88.66	93.16	90.50	**94.18**	90.19	92.65
Lamp	88.87	96.74	93.76	98.94	95.97	**98.27**	96.55	94.63
Phone	63.83	77.43	73.47	84.89	76.96	**87.25**	80.74	78.47
Mean	73.78	83.07	77.62	87.09	80.03	**90.18**	81.68	81.33
Stdev.	14.13	10.84	14.09	11.40	14.28	**7.31**	12.32	16.28

Table 3 shows 2DRE scores of EPro-PnP, FusionNet, and FusionNetV2. The employment of EBB also led to improvements in 2DRE scores, although the degree of improvement varied from object to object. The results were not significantly different from those of the ADD-0.1d score in Table 2. When EBBs were placed at Stage 1 and Stage 2, 2DRE scores showed the highest mean and the lowest variance. As a result, the 2DRE score was increased by 1.13 over FusionNet. However, the importance of placing EBB at Stage 2 seemed to be greater, and FusionNetV2 with one EBB at Stage 2 showed comparable (higher for several objects) performance to FusionNetV2 with EBBs at Stage 1 and Stage 2.

Table 3. Object-wise 2DRE scores of EPro-PnP, FusionNet, and FusionNetV2.

	EPro-PnP	FusionNet	FusionNetV2					
			s-1	s-2	s-3	s-1,2	s-1,2,3	s-1,2,3,4
Ape	97.52	98.00	97.81	**98.86**	97.81	98.48	98.00	98.38
Bench vise	96.70	95.64	95.64	96.99	96.31	**98.55**	96.51	97.67
Camera	95.10	98.43	97.75	**99.12**	98.04	**99.12**	97.75	97.94
Can	93.11	96.75	94.39	98.62	96.75	**99.02**	96.36	97.93
Cat	98.20	99.20	99.10	99.30	99.20	**99.40**	99.30	98.80
Driller	90.39	94.65	91.77	95.34	93.16	**96.23**	93.46	89.40
Duck	98.40	98.22	98.22	98.22	98.50	**98.69**	98.59	98.50
Egg box	98.59	99.06	98.87	**99.15**	98.69	**99.15**	98.69	98.97
Glue	93.63	97.88	95.95	**98.36**	96.53	98.17	95.56	97.68
Hole puncher	98.86	98.95	99.05	**99.81**	98.95	99.71	99.33	99.52
Iron	91.52	94.59	92.75	**95.81**	93.77	95.20	92.85	94.79
Lamp	90.88	94.82	91.36	**97.60**	93.57	97.12	94.72	94.72
Phone	92.26	96.32	93.77	97.17	95.56	**98.39**	96.69	95.94
Mean	95.01	97.12	95.88	98.03	96.68	**98.25**	96.75	96.94
Stdev.	3.19	1.74	2.83	1.37	2.12	**1.31**	2.13	2.73

Figure 1 shows the feature maps output from Stage 1 and Stage 2 of the CNN backbones of EPro-PnP, FusionNet, and FusionNetV2. The regions of brighter colors correspond to stronger features. The feature maps of EPro-PnP have bright regions irrelevant to object edges. FusionNet tends to boost features around object edges. However, compared to FusionNet, FusionNetV2 is shown to enhance features around object edges more effectively while strongly suppressing other features. This allowed FusionNetV2 to estimate 3D

coordinate maps more accurately (Figure 5), which is why FusionNetV2 outperforms EPro-PnP and FusionNet in 6D object pose estimation.

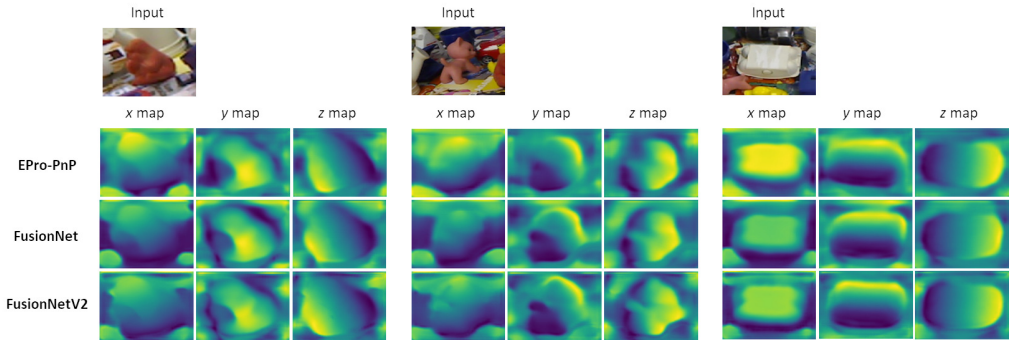

Figure 5. The 3D coordinate maps estimated by EPro-PnP, FusionNet, and FusionNetV2 for the same scenes. The object boundaries are most clearly seen in the results of FusionNetV2, indicating that the 3D coordinate maps are more precisely estimated.

6.3. Training Stability and Generalization Ability

In Figure 6, the training errors over the number of epochs of FusionNet and FusionNetV2 are visualized. The convergence speed of FusionNetV2 is slightly slower than that of FusionNet, and the training errors of FusionNetV2 begin to converge at around 100 epochs. However, it was found that the training errors decrease steadily and FusionNetV2 is stably trained with small errors when the number of epochs is more than 100. In addition, the validation accuracies of FusionNet and FusionNetV2 are visualized in Figure 7 to evaluate their generalization capabilities. Although verification accuracies increase steadily with each epoch number, the verification accuracy of FusionNetV2 is higher than that of FusionNet, which shows better generalization ability of FusionNetV2.

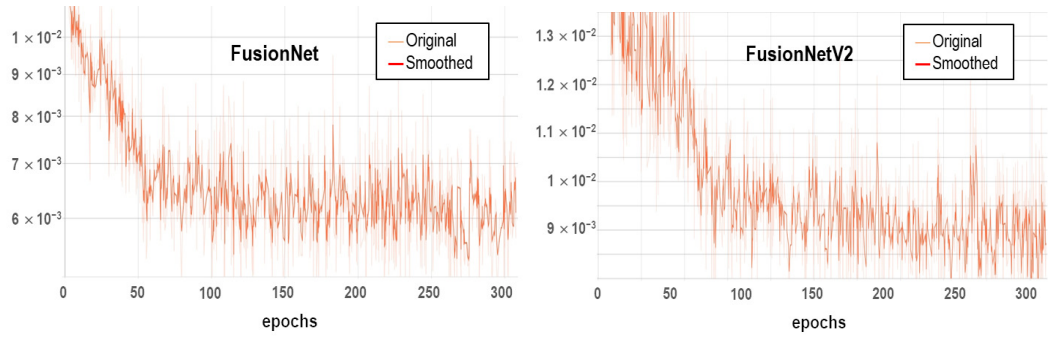

Figure 6. Training error/loss curves of FusionNet and FusionNetV2.

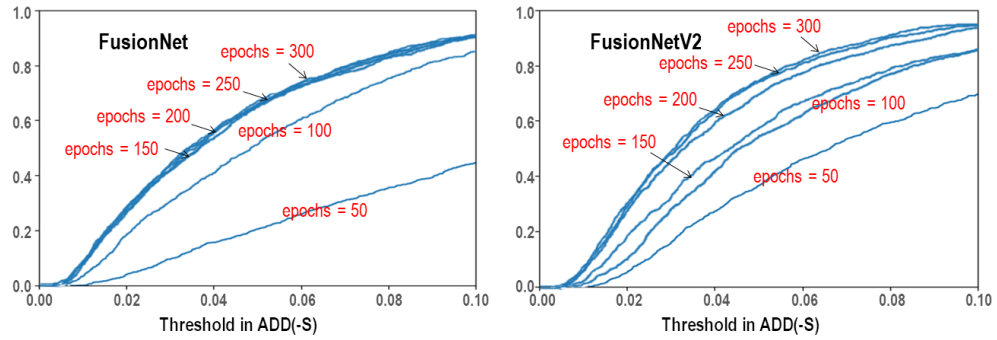

Figure 7. Validation accuracy curves of FusionNet and FusionNetV2.

6.4. Using Edge Features for 6D Object Pose Estimation

The core idea of FusionNetV2 is the enhancement of edge features, which has been demonstrated to bring performance improvements in 6D object pose estimation. Similar to FusionNetV2, ER-Pose [25] also noted the importance of edge features in a different framework. It obtained 3D-2D correspondences of keypoints using only semantic information on object edges and showed that improved accuracy can be achieved in 6D object pose estimation. Therefore, we attempted to adopt a similar strategy to ER-Pose and to find out if it is valid in our framework. In this regard, we simply extracted edge maps from input images and used them as masks to filter dense 3D coordinate maps in the EPro-PnP head. The ADD-0.1d and 2DRE results are shown in Figure 8 and are rather less accurate for all objects. As a result, the approach, which uses only 3D coordinates on object edges for 6D object pose estimation, is shown to be unfavorable in the framework of EPro-PnP, FusionNet, and FusionNetV2. In fact, this was predictable from the results in Section 6.2. As discussed, the loss of semantic (or high-level) information such as 3D coordinates results in compromising accuracy. In contrast, our approach to enhancing shallow edge features has proven effective.

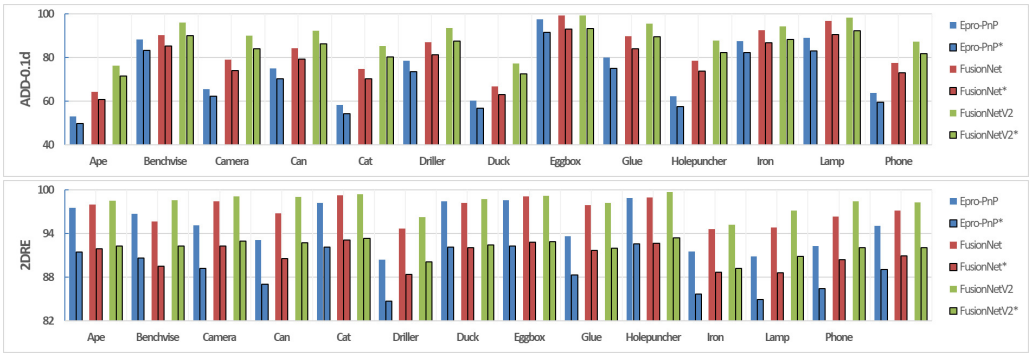

Figure 8. Effects of filtering 3D coordinate maps using edge maps. * indicates the result of filtering being applied.

6.5. Inference Time

The EBB is light in weight and has little effect on the inference time of FusionNetV2. Furthermore, all AtBs of FusionNet were eliminated, allowing FusionNetV2 to be faster than FusionNet. As shown in Figure 9, the inference time of FusionNetV2 was 0.017 ms, slightly shorter than that of FusionNet. The small reduction in inference time is due to AtB being lightweight, consisting only of 1×1 convolution layers.

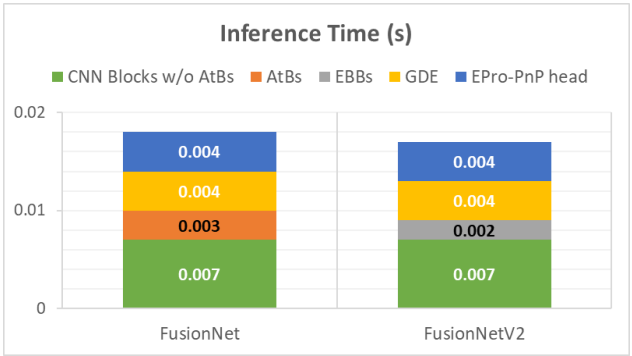

Figure 9. Inference time.

In order to further reduce the inference time of FusionNetV2, we attempted to improve the GDE block of a three-layer Transformer structure. Transformers are essentially composed of multiple layers of self-attention. Therefore, model complexity increases quadratically with the length of the input sequence. Taking this into account, we con-

ducted experiments to verify the performance of the linear Transformers (LTs) [42] and fast feedforward (FFF) networks [43], which are known to improve the inference speed of Transformers.

First, we attempted to replace the Transformers in the GDE block with LTs while keeping the depth at three layers. To ensure a fair comparison, the original structure of FusionNetV2 was retrained and only the GDE part was modified. The results are shown in Table 4. It is shown that the inference time can be slightly reduced by using LT. However, using LT resulted in a considerable drop in the ADD 0.1d score. Second, we attempted to replace the Transformer's feedforward network with FFF. However, the replacement with FFF rather greatly increased the inference time. Nevertheless, the accuracy dropped significantly. This tendency of LT and FFF was the same with FusionNet. Consequently, the application of LT or FFF to FusionNet and FusionNetV2 is not recommended, except for applications that require extremely high inference-time efficiency.

Table 4. Ablation experiments for LT and FFF.

	FFF	LT	Time [1] (s)	ADD-0.1d
FusionNet	-	-	0.004	83.07
	√	-	0.009	81.8
	-	√	0.003	81.05
FusionNetV2	-	-	0.004	90.18
	√	-	0.009	82.58
	-	√	0.003	81.69

[1] It is the time spent on the GDE part only.

7. Conclusions

In this study, we introduced FusionNetV2, an improved version of FusionNet. Focusing on the fact that FusionNet tends to enhance features around object edges, we attempted to explicitly boost features on object edges, and for this purpose, an edge attention block named EBB was proposed. FusionNetV2 was designed to have the same architecture as FusionNet, except for removing AtBs and adding two EBBs. Through experiments, the impact of placing EBBs at different stages of FusionNetV2 was analyzed and it was demonstrated that FusionNetV2 with EBBs at the end of Stage 1 and Stage 2 enhanced edge features most effectively, achieving significantly higher accuracy than EPro-PnP and FusionNet of the same framework in 6D object pose estimation. As a result, it was confirmed that explicitly enhancing the features around object edges contributes significantly to improving the performance of 6D object pose estimation. Furthermore, the simple and light EBB allowed FusionNetV2 to be faster in inference than FusionNet. However, incorrect EBB placement reduced pose estimation accuracy, and methods used in other frameworks to enhance edge features for pose estimation or to improve the inference speed of Transformers were not available in our framework. This indicates the need for more sophisticated methods to improve edge features and improve inference speed, which needs to be addressed in future studies.

Unfortunately, our approach to explicitly boosting edge features can be inherently vulnerable to occlusion or clutter. Therefore, it would be an interesting future study to analyze the performance of FusionNetV2 against occlusion or clutter and to find ways to improve its performance.

In addition, the network structure is one of the important factors influencing the performance of 6D pose estimation. Therefore, finding a better network structure for our framework would be another interesting area for future research.

Author Contributions: Conceptualization, Y.Y. and H.P.; Funding acquisition, H.P.; Methodology, Y.Y. and H.P.; Software, Y.Y.; Supervision, H.P.; Validation, Y.Y. and H.P.; Writing—original draft, Y.Y.; Writing—review and editing, H.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant by the Korean Government through the MSIT under grant 2021R1F1A1045749.

Data Availability Statement: The data that support the findings of this study are publicly available in the online repository: https://bop.felk.cvut.cz/datasets/ (accessed on 21 August 2024).

Conflicts of Interest: We have no conflicts of interest to declare.

Abbreviations

The following abbreviations are used in this manuscript:

PnP	perspective-n-point
CNN	convolutional neural network
NLP	natural language processing
ViT	Vision Transformer
AtB	attention block
EBB	edge boosting block
GDE	global dependency encoder
BN	batch normalization
2DRE	2D reprojection error
LT	linear Transformer
FFF	fast feedforward network

References

1. Lepetit, V.; Moreno-Noguer, F.; Fua, P. EPnP: An accurate O(n) solution to the PnP problem. *Int. J. Comput. Vis.* **2009**, *81*, 155–166. [CrossRef]
2. Hosang, J.; Benenson, R.; Schiele, B. Learning non-maximum suppression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4507–4515.
3. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
4. Hu, H.; Gu, J.; Zhang, Z.; Dai, J.; Wei, Y. Relation networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3588–3597.
5. Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; Agrawal, A. Context encoding for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7151–7160.
6. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
7. Cao, Y.; Xu, J.; Lin, S.; Wei, F.; Hu, H. GCNet: Non-local networks meet squeeze-excitation networks and beyond. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019; pp. 1971–1980.
8. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010.
9. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv* **2018**, arXiv:1804.07461.
10. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2010**, arXiv:2010.11929.
11. Hatamizadeh, A.; Yin, H.; Heinrich, G.; Kautz, J.; Molchanov, P. Global context vision transformers. In Proceedings of the International Conference on Machine Learning, 2023, ICML'23, Honolulu, HI, USA, 23–29 July 2023.
12. Wu, H.; Xiao, B.; Codella, N.; Liu, M.; Dai, X.; Yuan, L.; Zhang, L. CvT: Introducing Convolutions to Vision Transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual Conference, 11–17 October 2021; pp. 22–31. [CrossRef]
13. Ye, Y.; Park, H. FusionNet: An End-to-End Hybrid Model for 6D Object Pose Estimation. *Electronics* **2023**, *12*, 4162. [CrossRef]
14. Chen, H.; Wang, P.; Wang, F.; Tian, W.; Xiong, L.; Li, H. EPro-PnP: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 2781–2790.
15. Wang, Y.; Jiang, X.; Fujita, H.; Fang, Z.; Qiu, X.; Chen, J. EFN6D: An efficient RGB-D fusion network for 6D pose estimation. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *15*, 75–88. [CrossRef]

16. Dam, T.; Dharavath, S.B.; Alam, S.; Lilith, N.; Chakraborty, S.; Feroskhan, M. AYDIV: Adaptable Yielding 3D Object Detection via Integrated Contextual Vision Transformer. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 3–17 May 2024; pp. 10657–10664.
17. Rad, M.; Lepetit, V. BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3828–3836.
18. Peng, S.; Liu, Y.; Huang, Q.; Zhou, X.; Bao, H. PVNet: Pixel-wise voting network for 6DoF pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4561–4570.
19. Pavlakos, G.; Zhou, X.; Chan, A.; Derpanis, K.G.; Daniilidis, K. 6-DoF object pose from semantic keypoints. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–June 2017; pp. 2011–2018.
20. Zhao, Z.; Peng, G.; Wang, H.; Fang, H.; Li, C.; Lu, C. Estimating 6D pose from localizing designated surface keypoints. *arXiv* **2018**, arXiv:1812.01387.
21. Ullah, F.; Wei, W.; Daradkeh, Y.I.; Javed, M.; Rabbi, I.; Al Juaid, H. A Robust Convolutional Neural Network for 6D Object Pose Estimation from RGB Image with Distance Regularization Voting Loss. *Sci. Program.* **2022**, *2022*, 2037141. [CrossRef]
22. Oberweger, M.; Rad, M.; Lepetit, V. Making deep heatmaps robust to partial occlusions for 3D object pose estimation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 119–134.
23. Haugaard, R.L.; Buch, A.G. SurfEmb: Dense and Continuous Correspondence Distributions for Object Pose Estimation with Learnt Surface Embeddings. *arXiv* **2021**, arXiv:2111.13489.
24. Hai, Y.; Song, R.; Li, J.; Ferstl, D.; Hu, Y. Pseudo Flow Consistency for Self-Supervised 6D Object Pose Estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–3 October 2023; pp. 14075–14085.
25. Yang, X.; Li, K.; Wang, J.; Fan, X. ER-Pose: Learning edge representation for 6D pose estimation of texture-less objects. *Neurocomputing* **2023**, *515*, 13–25. [CrossRef]
26. Li, F.; Vutukur, S.R.; Yu, H.; Shugurov, I.; Busam, B.; Yang, S.; Ilic, S. NeRF-Pose: A first-reconstruct-then-regress approach for weakly-supervised 6D object pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–3 October 2023; pp. 2123–2133.
27. Wu, Y.; Greenspan, M. Learning Better Keypoints for Multi-Object 6DoF Pose Estimation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 1–6 January 2024; pp. 564–574.
28. Jantos, T.G.; Hamdad, M.A.; Granig, W.; Weiss, S.; Steinbrener, J. PoET: Pose Estimation Transformer for Single-View, Multi-Object 6D Pose Estimation. In Proceedings of the Conference on Robot Learning. PMLR, Atlanta, GA, USA, 6–9 November 2023; pp. 1060–1070.
29. Periyasamy, A.S.; Amini, A.; Tsaturyan, V.; Behnke, S. YOLOPose V2: Understanding and improving transformer-based 6D pose estimation. *Robot. Auton. Syst.* **2023**, *168*, 104490. [CrossRef]
30. Zhang, Z.; Chen, W.; Zheng, L.; Leonardis, A.; Chang, H.J. Trans6D: Transformer-Based 6D Object Pose Estimation and Refinement. In *Computer Vision—ECCV 2022 Workshops*; Karlinsky, L., Michaeli, T., Nishino, K., Eds.; Springer: Cham, Switzerland, 2023; pp. 112–128.
31. Castro, P.; Kim, T.K. CRT-6D: Fast 6D object pose estimation with cascaded refinement transformers. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 2–7 January 2023; pp. 5746–5755.
32. Wen, B.; Yang, W.; Kautz, J.; Birchfield, S. FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects. *arXiv* **2023**, arXiv:2312.08344.
33. Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Gool, L.V.; Timofte, R. SwinIR: Image Restoration Using Swin Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Virtual Conference, 11–17 October 2021; pp. 1833–1844. [CrossRef]
34. Li, X.; Xiang, Y.; Li, S. Combining convolutional and vision transformer structures for sheep face recognition. *Comput. Electron. Agric.* **2023**, *205*, 107651. [CrossRef]
35. He, L.; He, L.; Peng, L. CFormerFaceNet: Efficient Lightweight Network Merging a CNN and Transformer for Face Recognition. *Appl. Sci.* **2023**, *13*, 6506. [CrossRef]
36. Mogan, J.N.; Lee, C.P.; Lim, K.M.; Ali, M.; Alqahtani, A. Gait-CNN-ViT: Multi-Model Gait Recognition with Convolutional Neural Networks and Vision Transformer. *Sensors* **2023**, *23*, 3809. [CrossRef] [PubMed]
37. Lin, Y.; Zhang, D.; Fang, X.; Chen, Y.; Cheng, K.T.; Chen, H. Rethinking Boundary Detection in Deep Learning Models for Medical Image Segmentation. In *International Conference on Information Processing in Medical Imaging*; Springer: Cham, Switzerland, 2023; pp. 730–742.
38. Kanopoulos, N.; Vasanthavada, N.; Baker, R.L. Design of an image edge detection filter using the Sobel operator. *IEEE J. Solid-State Circuits* **1988**, *23*, 358–367. [CrossRef]
39. Hinterstoisser, S.; Cagniart, C.; Ilic, S.; Sturm, P.; Navab, N.; Fua, P.; Lepetit, V. Gradient response maps for real-time detection of textureless objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 876–888. [CrossRef] [PubMed]
40. Brachmann, E.; Michel, F.; Krull, A.; Yang, M.Y.; Gumhold, S. Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26–27 June 2016; pp. 3364–3372.

41. Li, Z.; Wang, G.; Ji, X. CDPN: Coordinates-based disentangled pose network for real-time RGB-based 6-DoF object pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7678–7687.
42. Katharopoulos, A.; Vyas, A.; Pappas, N.; Fleuret, F. Transformers are RNNs: Fast autoregressive transformers with linear attention. In Proceedings of the International Conference on Machine Learning. PMLR, Virtual Event, 13–18 July 2020; pp. 5156–5165.
43. Belcak, P.; Wattenhofer, R. Fast feedforward networks. *arXiv* **2023**, arXiv:2308.14711.