actuators

MDPI

Article

Robot Closed-Loop Grasping Based on Deep Visual Servoing Feature Network

Junqi Luo [1], Zhen Zhang [2,*], Yuangan Wang [1,*] and Ruiyang Feng [1]

[1] School of Electric and Information Engineering, Beibu Gulf University, Qinzhou 535000, China; luo.junqi@bbgu.edu.cn (J.L.); ruiyang.feng@hotmail.com (R.F.)
[2] School of Electronic Information and Automation, Guilin University of Aerospace Technology, Guilin 541000, China
* Correspondence: 2023004@guat.edu.cn (Z.Z.); ygw@bbgu.edu.cn (Y.W.)

Abstract: Robot visual servoing for grasping has long been challenging to execute in complex visual environments because of issues with efficient feature extraction. This paper proposes a novel visual servoing grasping approach based on the Deep Visual Servoing Feature Network (DVSFN) to tackle this issue. The approach enables feasible to extract scale-invariant point features and target bounding boxes in real time by building an effective single-stage multi-dimensional feature extractor. The DVSFN is then integrated into a Levenberg–Marquardt–based image visual servoing (LM-IBVS) controller. The above creates a mapping link between the robot's joint space and image features. The robot is then guided in positioning and grabbing by converting the difference between the expected and present features into the corresponding robot joint velocities. Experimental results demonstrate that the proposed method achieves a mean average precision (mAP) of 0.80 and 0.87 for detecting target bounding boxes and point features, respectively, in scenarios with significant lighting variations and occlusions. Under low-light and partial occlusion conditions, the method achieves an average grasping success rate approximately 80%.

Keywords: robot grasping; visual servoing; object detection; complex visual environments

check for updates

Received: 6 December 2024
Revised: 28 December 2024
Accepted: 9 January 2025
Published: 12 January 2025

Citation: Luo, J.; Zhang, Z.; Wang, Y.; Feng, R. Robot Closed-Loop Grasping Based on Deep Visual Servoing Feature Network. *Actuators* **2025**, *14*, 25. https://doi.org/10.3390/act14010025

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Robotic grasping techniques are essential to contemporary production, providing automation and accuracy for intricate jobs. Visual servoing integrates visual sensor feedback directly into the robot's control loop, enabling real-time posture adjustments based on current visual features [1]. This strategy is far more flexible than conventional "look-then-move" tactics. However, despite much research, using robot visual servoing in complicated visual situations is still quite challenging [2].

Visual servoing for grasping is typically categorized into position-based visual servoing (PBVS) and image-based visual servoing (IBVS) [3]. PBVS methods rely on the target's 3D model to estimate target poses, which can be computationally intensive. Conversely, IBVS avoids explicit 3D reconstruction but often depends on traditional image processing techniques, such as SIFT or ARUCO markers [4]. These methods struggle to ensure robust feature extraction in visually complex conditions, limiting their effectiveness [5,6].

Recent advancements in robot grasping research have increasingly focused on deep learning, encompassing planar and 6-DoF (degrees of freedom) grasping. Planar grasping involves selecting an optimal 4D grasp configuration from a set of candidates, while 6-DoF grasping determines the best solution from six-dimensional regression results. Despite their progress, these paradigms face challenges: planar grasping keypoint information varies

significantly with viewpoint changes, while 6-DoF grasping's high computational demands (inference times ranging from hundreds of milliseconds to several seconds) hinder its integration into real-time visual servoing loops.

This paper proposes a novel integration of deep learning techniques with domain knowledge in robot visual servoing to address these limitations. We develop a keypoint detection network tailored for robotic visual servoing grasping designed to overcome the challenges of real-time, robust feature extraction in complex visual conditions. Unlike traditional deep learning networks, the proposed approach detects both target bounding boxes and scale-invariant keypoints, seamlessly combining these features with an IBVS framework to enable precise object localization and tracking. This method bridges the gap between learning-based techniques and visual servoing systems. The main contributions of this paper are as follows:

(1) We present a single-stage, multi-dimensional feature extraction network inspired by the efficient backbone architectures of the YOLO series. The redesigned network head enables simultaneous extraction of bounding boxes and keypoints, providing inherent compatibility with visual servoing systems;
(2) A novel control framework is proposed to map the image features extracted by DVSFN directly to the robot's joint space, facilitating accurate guidance of the robot to its target;
(3) Extensive experiments validate the proposed method, demonstrating high detection accuracy and robust grasping performance in visually complex environments.

This paper is organized as follows: Section 3 provides a detailed discussion of the proposed method's architecture, including the visual servoing feature detection network and the LM-IBVS framework. Section 4 presents a series of experiments to evaluate and validate the performance of the proposed networks. Section 5 outlines the results and analysis of physical grasping experiments. Finally, Section 6 concludes this paper.

2. Related Works

Current research in visual servoing for robotic grasping frequently presumes that cameras can reliably extract image features. However, challenges such as feature interference and loss due to complex visual backgrounds and occlusions—common in real-world scenarios—have received limited attention. Additionally, the speed of image processing remains a critical bottleneck for the practical deployment of visual servoing in grasping tasks. Consequently, beyond constructing vision-based visual servoing frameworks, achieving robust and real-time extraction of target image features is vital for reliable performance.

In visual servoing, the primary goal of feature detection is to extract scale-invariant features. Huang et al. [7] utilized improved Harris and SIFT algorithms for feature point extraction. To address feature mismatches, Huang et al. [8] integrated the ORB (Oriented FAST and Rotated BRIEF) algorithm with Maximum Likelihood Estimation Random Sample Consensus (MLESAC), enhancing the accuracy of feature extraction and matching. With the advent of deep learning, learning-based approaches have gained prominence. Guo et al. [9] employed convolutional neural networks (CNNs) to detect target bounding boxes and combined them with image-based visual servoing (IBVS) for real-time tracking. Liu et al. [5] refined the YOLOv3 algorithm for detecting and recognizing objects, using the bounding box corners as features to guide robotic localization tasks. However, as bounding boxes lack rotational information, IBVS methods based on such features are better suited to tracking static or translational targets.

Several studies [10–12] propose end-to-end visual servoing strategies that extract features from current and target image pairs directly. These approaches leverage feature extraction and convolutional layers to generate large-scale feature maps, followed by fully connected layers for translational and rotational regression. While bypassing

explicit feature extraction, these methods require retraining to recalibrate scene/camera pose mappings whenever the hand–eye configuration changes. Harish et al. [13] integrated optical flow and depth estimators into IBVS controllers to extract visual and depth features, albeit at high computational costs due to pixel-wise predictions. Adrian et al. [14] employed a self-supervised shared encoder to detect keypoints and incorporated them into an IBVS controller automatically. Despite its effectiveness, this method demands significant computational resources for outlier rejection and post-processing.

In summary, traditional image processing and feature matching methods dominate image feature detection in visual servoing. However, these methods struggle with challenges such as target occlusion, poor lighting, and cluttered scenes, which are prevalent in robotic grasping tasks. Deep learning-based methods perform better in unstructured environments but face significant hurdles, including dataset construction, inference speed, and integration with visual servoing systems. Key challenges include the following: (1) Overly simplistic feature representations. Standard grasping datasets, such as Cornell Grasp [15], Jacquard [16], and GraspNet-1Billion [17], typically employ grasping box representations that are ill-suited for visual servoing. (2) Real-time requirements and the need for accurate, scale-invariant feature extraction, which existing algorithms often fail to meet simultaneously. (3) Robustness in complex visual conditions, such as low light, occlusion, and clutter, which remains under-explored in current visual servoing research [18,19].

3. Proposed Method

3.1. Design of the DVSFN

The proposed Deep Visual Servoing Feature Network (DVSFN) adopts a YOLO-like architecture, as illustrated in Figure 1. The backbone leverages Darknet-53 to process the last three feature maps through a hybrid encoder, generating spatial features P4 and P5 with downsampling rates of 16 and 32, respectively. Each pixel in these feature maps corresponds to a uniformly distributed grid on the original image plane. The network head incorporates depth-wise separable convolution (DSC) for channel-wise feature extraction, followed by point-wise convolution for information fusion. Each grid unit predicts the score, bounding box, keypoint coordinates, and visibility features.

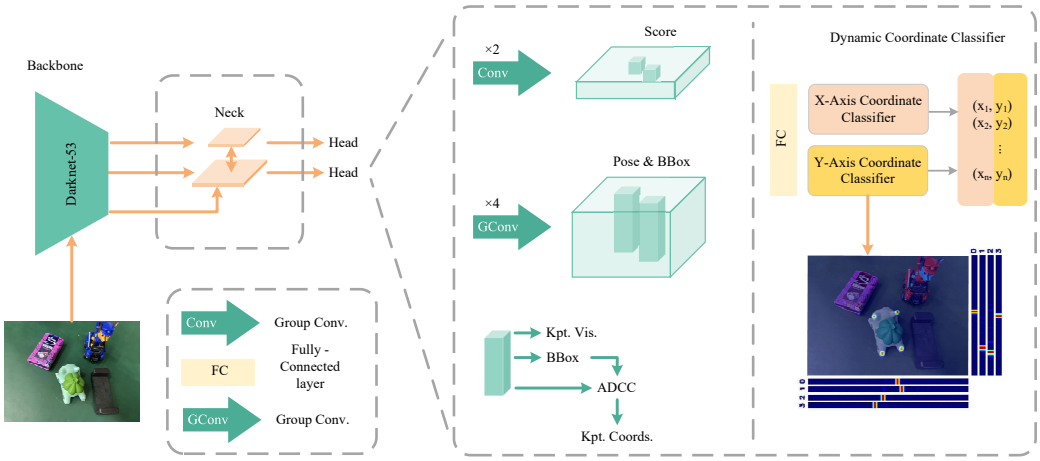


Figure 1. Schematic diagram of DVSFN architecture.

Adaptive Dynamic Coordinate Classifier. The displacement of keypoints relative to each grid unit is a crucial pose feature. Traditional methods predict these displacements through direct regression, which is computationally intensive [20,21]. The SIMCC method [22] mitigates this issue by employing fixed interval counts (for example, Bx = 192, By = 256). However, this approach is inefficient for small targets and insufficiently precise

for large targets. To overcome these limitations, we propose an adaptive interval allocation strategy that adjusts interval density dynamically using the following formula:

$$B_{\text{adaptive}} = \left\lceil s \cdot \sqrt{\frac{w \cdot h}{\alpha}} \right\rceil \tag{1}$$

where w and h are the width and height of the target bounding box, s is a scaling factor (default s = 1.5), and α controls interval density. This strategy optimizes computational efficiency by dynamically scaling interval counts based on target size, ensuring precision for larger targets while minimizing resource waste for smaller ones.

Gaussian Function Position Encoding. We enhance interval representation by employing Gaussian function position encoding, which captures local spatial relationships more effectively than sinusoidal encodings. Each position is encoded through the application of a symmetric Gaussian distribution, as follows:

$$\text{PE}_{\text{gaussian}}\left(x_i\right)_c = \exp\left(-\frac{\left(x_i - \mu_c\right)^2}{2\sigma_c^2}\right) \tag{2}$$

where μ_c is the mean, σ_c^2 is the standard deviation (initially 0.1), and x_i is the position index. This encoding prioritizes local variations, making it well suited for targets in specific regions.

Sparse Computation and Attention Mechanism. To reduce computational overhead, we incorporate a sparse attention mechanism. Traditional attention mechanisms process all positions, which can be computationally expensive. Sparse attention retains only the k most relevant position encodings for the current query:

$$\text{SparseAttention}(Q, K, V) = \text{TopK}(Q, k) \cdot V \tag{3}$$

where Q is the query matrix, K is the key matrix, and V is the value matrix.

Probability Prediction. Building upon the work in [23], we refine probability predictions using an adaptive temperature and feature normalization mechanism:

$$\hat{p}_k(x_i) = \sigma\left(\frac{F_k \cdot \phi(\text{PE}_{\text{multi}}\left(x_i\right))}{\tau(F_k)}\right) \tag{4}$$

where $\tau(F_k)$ is a learnable temperature function, and ϕ applies a non-linear transformation (using GELU) for improved accuracy and robustness.

Maximum Likelihood Estimation for Coordinate Classification. In classification tasks, a one-hot target and a cross-entropy loss function are customarily utilized. The label smoothing technique and maximum likelihood estimation (MLE) employed in SIMCC can improve performance. To enhance model accuracy and stability, we propose the Adaptive Maximum Likelihood Estimation (A-MLE) loss, which introduces adaptive weights and learned variance to improve prediction accuracy under uncertain visual conditions further. The A-MLE loss function is defined as

$$L_{\text{A-MLE}} = -\log\left[\sum_{i=1}^{B_{\text{adaptive}}} \omega_i \cdot \frac{1}{\hat{\sigma}} \exp\left(-\frac{|x_i - \mu_x|^2}{2\hat{\sigma}^2 s}\right) \hat{p}_k(x_i)\right] \tag{5}$$

where ω_i is the adaptive weight, $\hat{\sigma}$ represents predicted variance, and s normalizes by the instance scale. These parameters are initialized as 1 and dynamically adjusted based on the sample error or model confidence.

Loss Function Design. The total loss function is devised to optimize multiple objectives: bounding box prediction, keypoint localization, classification, and temporal consistency. It incorporates several loss components: the bounding box IoU loss L_{bbox} for evaluating the overlap between predicted and ground truth boxes; the adaptive MLE loss $L_{\text{A-MLE}}$ for enhancing keypoint prediction accuracy; the proxy loss L_{proxy} based on OKS to assess keypoint similarity; the cross-entropy classification loss L_{cls} for optimizing target classification; the binary cross-entropy loss L_{vis} for keypoint visibility estimation; and the consistency loss $L_{\text{consistency}}$ to ensure stability across temporal frames. The total loss function is as follows:

$$L_{\text{total}} = \lambda_1 \cdot L_{\text{bbox}} + \lambda_2 \cdot L_{\text{A-MLE}} + \lambda_3 \cdot L_{\text{proxy}} + \lambda_4 \cdot L_{\text{cls}} + L_{\text{vis}} + \lambda_5 \cdot L_{\text{consistency}} \quad (6)$$

where λ_1, λ_2, λ_3 and λ_4 are the weight coefficients for each loss term. In this paper, we set $\lambda_1 = \lambda_2 = 5$, $\lambda_3 = 10$, $\lambda_4 = 2$.

3.2. Design of the LM-IBVS

This section outlines the integration of keypoint coordinates, extracted from DVSFN, with the IBVS framework to form a closed-loop grasping controller. The controller's objective is to minimize the pixel discrepancy between the target and the current image feature points, thereby enabling the generation of real-time velocity control commands for the robot's six joints. The structural block diagram of the LM-IBVS algorithm is shown in Figure 2.

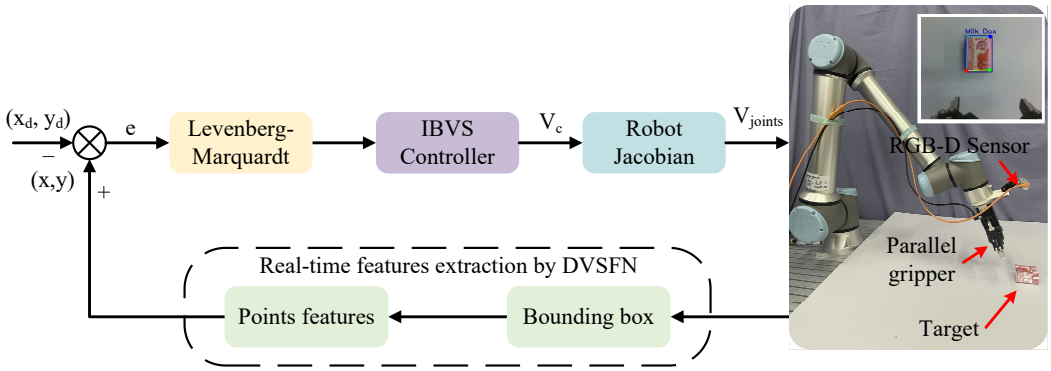


Figure 2. Structural block diagram of the LM-IBVS controller.

While the fundamental structure of the IBVS controller remains unchanged, it is adapted to fit the LM controller framework. In IBVS, the image Jacobian matrix describes the relationship between feature point velocities and camera velocities, as formulated below.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -\frac{f}{Z} & 0 & \frac{x}{Z} & \frac{xy}{f} - \frac{f^2+x^2}{f} & y \\ 0 & -\frac{f}{Z} & \frac{f}{Z} & \frac{x^2+y^2}{f} - \frac{xy}{f} & -x \end{bmatrix} \begin{bmatrix} T_{cX} \\ T_{cY} \\ T_{cZ} \\ \Omega_{cX} \\ \Omega_{cY} \\ \Omega_{cZ} \end{bmatrix} \quad (7)$$

where f is the camera focal length, Z denotes the depth of the feature point, $T_c = (T_{cX}, T_{cX}, T_{cX})$ represents the camera's linear velocity, and $\Omega_c = (\Omega_{cX}, \Omega_{cY}, \Omega_{cZ})$ corresponds to the camera's angular velocity.

We introduce an error term, defining the image error as

$$e = \begin{bmatrix} x_d - x \\ y_d - y \end{bmatrix} \tag{8}$$

where (x_d, y_d) represents the desired feature point, and (x, y) denotes the current feature point.

In robotic grasping tasks, the IBVS control is essentially a process of minimizing the image error cost function, expressed as

$$\dot{e} = J_{\text{image}} \, \Delta u \tag{9}$$

where J_{image} is the image Jacobian matrix, $\dot{e}$ represents the rate of change in image error, and Δu is the camera velocity.

By incorporating the LM algorithm, the LM-IBVS control law is derived as follows:

$$\begin{bmatrix} T_{cX} \\ T_{cY} \\ T_{cZ} \\ \Omega_{cX} \\ \Omega_{cY} \\ \Omega_{cZ} \end{bmatrix} = -\left(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I}\right)^{-1}\mathbf{J}^T \begin{bmatrix} x_d - x \\ y_d - y \end{bmatrix} \tag{10}$$

where $\mathbf{J}^T\mathbf{J}$ approximates the Hessian matrix in the Gauss–Newton method, λ is the damping factor of the LM method, and $\mathbf{I}$ is the identity matrix.

4. Training and Evaluation of DVSFN

4.1. Datasets and Training Strategies

The proposed model was evaluated using a vision-based servo grasping dataset annotated by our team. This dataset includes 1200 manually labeled real-world images across six categories and 9288 synthetic images. We employed domain randomization techniques using Unity3D [24] to simulate various environmental factors, including occlusion, lighting variations, and scale changes. This significantly enhances the diversity of the dataset, improving the model's ability to generalize across different conditions. Finally, we trained the model on a mixture of both synthetic and real-world samples, which effectively bridges the gap between simulated and real-world environments and helps mitigate performance degradation that may arise from relying solely on synthetic data. Annotations for real and synthetic samples were standardized to ensure consistency. The synthetic dataset and 65% of the real-world samples were used for training, while the remaining real samples were reserved for testing.

The training was carried out over 200 epochs using the AdamW optimizer, with an initial learning rate of 0.004, the optimizer momentum set at 0.8, a batch size of 64, and 300 warm-up iterations. To mitigate overfitting, a weight decay of 0.05 was applied. A step decay learning rate schedule maintained a constant learning rate during most training iterations. It progressively reduced logarithmically from epoch 150 onwards, ensuring smooth convergence and minimizing instability in later stages.

4.2. Evaluation Metric

Model performance was evaluated concerning both object localization and keypoint localization tasks. Standard COCO metrics—AP50 and mAP—were used for object local-

ization. Keypoint localization accuracy was quantified using the Object Keypoint Similarity (OKS) metric, defined as

$$OKS = \frac{\sum_i \exp\left(-d_i^2/2s^2j_i^2\right)\sigma(v_i > 0)}{\sum_i \sigma(v_i > 0)} \tag{11}$$

where d_i is the Euclidean distance between the i-th detected keypoint and its ground truth, while v_i denotes the visibility status of the i-th keypoint. Precisely, $v_i = 0$ corresponds to an unannotated point, $v_i = 1$ indicates an annotated but invisible point, and $v_i = 2$ signifies an annotated and visible point. The function is an indicator function. Additionally, j_i is the standard deviation for the i-th keypoint, set to 0.01 in this paper, and s denotes the object scale, automatically computed based on the bounding box dimensions in the image.

To assess the detection accuracy under a specific threshold T, the average precision (AP) is defined as

$$AP = \frac{\sum_p \delta\left(OKS_p > \mathrm{T}\right)}{\sum_p l} \tag{12}$$

where $\delta\left(OKS_p > \mathrm{T}\right)$ indicates whether the OKS of a detected keypoint p exceeds the threshold T, and l denotes the total number of keypoints.

To provide a comprehensive evaluation of the model's performance across varying precision requirements, the mean average precision (mAP) metric is introduced:

$$mAP = \mathrm{mean}[AP@(0.50 : 0.05 : 0.95)] \tag{13}$$

This metric calculates the mean AP across thresholds ranging from 0.5 to 0.95, with a step size 0.05. A higher mAP value reflects superior overall performance in keypoint localization.

4.3. Quantitative Analysis of Experimental Results

Within the DVSFN framework, we evaluated the performance of four backbone networks, namely R-50-FPN, R-101-FPN, CSPNeXt, and Darknet-53. As summarized in Table 1, Darknet-53 achieves superior performance across all metrics. Compared to R-101-FPN, Darknet-53 improves box mAP by 3.21% and keypoint mAP by 4.19%. Against the next-best backbone, CSPNeXt, it achieves a 1.51% increase in box mAP (from 0.793 to 0.805), a 1.75% boost in keypoint mAP (from 0.855 to 0.870), and a 1.11% gain in keypoint AP50, while offering faster inference at 39 FPS compared to CSPNeXt's 34 FPS. These results highlight Darknet-53's balance between accuracy and real-time performance.

Table 1. Performance comparison of backbone networks.

Backbone	Box	Box	Keypoint	Keypoint	Inf. Time
	mAP	**AP50**	**mAP**	**AP50**	**FPS**
R-50-FPN	0.764	0.936	0.820	0.870	29
R-101-FPN	0.780	0.948	0.835	0.885	23
CSPNeXt	0.793	0.954	0.855	0.900	34
Darknet-53	**0.805**	**0.960**	**0.870**	**0.910**	**39**

The networks R-50-FPN and R-101-FPN, based on the ResNet architecture, help mitigate the gradient vanishing problem in deep networks. Additionally, the feature pyramid network (FPN) structure improves the extraction of multi-scale features. However, the FPN introduces additional computational overhead. In contrast, CSPNeXt adopts the design principles of CSPNet (cross-stage partial network), which divides the network into multiple stages to enhance performance. CSPNeXt employs both residual and cross-stage connections to facilitate multi-scale feature extraction, improving information flow efficiency

compared to traditional ResNet architectures. Darknet-53 combines bottleneck blocks and residual blocks, further enhanced by Leaky ReLU activation functions, significantly increasing the network's expressive power. Consequently, Darknet-53 is designed with a strong emphasis on computational efficiency and real-time performance, making it well suited for tasks such as visual servoing and robotic grasping, where both speed and accuracy are essential.

To further validate the proposed model, we compared it against three state-of-the-art detectors: Keypoint RCNN [25], HRNet [26], and RTMPose [27]. HRNet and RTMPose adopt two-stage designs, integrating YOLOvX for object detection. For consistency, all models were trained on the same dataset for 200 epochs, and inference efficiency was evaluated based on their average FPS over 30 s. The results are shown in Figure 3.

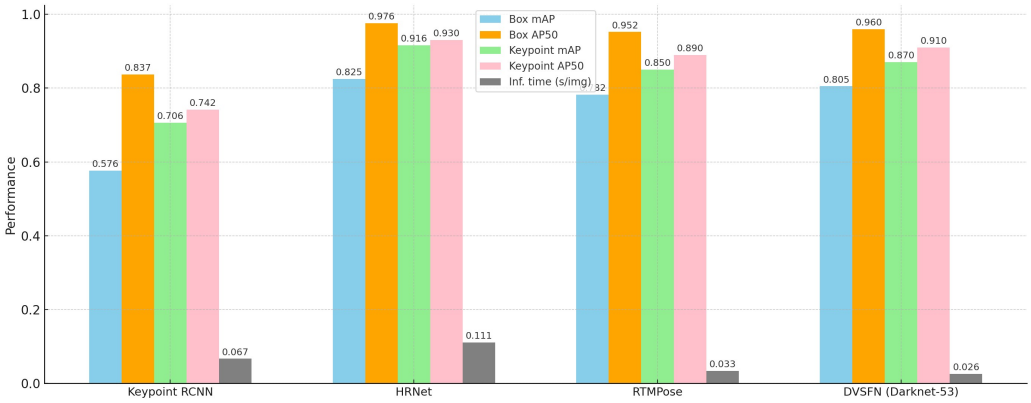


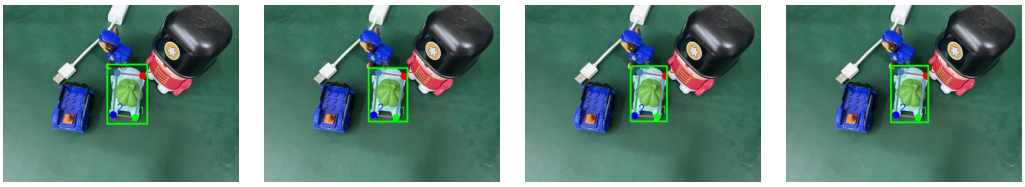
Figure 3. Model performance comparison.

The proposed method achieves a box mAP of 0.805 and a box AP50 of 0.960, outperforming Keypoint RCNN by 0.229 and 0.123, respectively, and exceeding RTMPose in box mAP by 2.9%. HRNet delivers the highest accuracy in keypoint detection, with a keypoint mAP of 0.916 and an AP50 of 0.930. However, it suffers from low inference speed at 9 FPS, limiting its utility in real-time applications. RTMPose balances detection accuracy (keypoint mAP = 0.850) and inference speed (30 FPS).

The proposed model, utilizing Darknet-53, achieves a keypoint mAP that is just 2.15% lower than HRNet while significantly outperforming other models. Additionally, it achieves the highest inference speed at 39 FPS, 3.3 times faster than HRNet, demonstrating a more favorable balance between precision and efficiency.

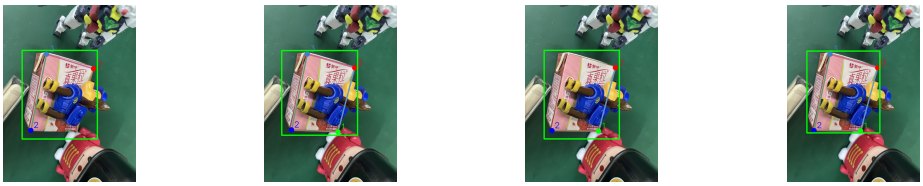
4.4. Qualitative Analysis of Experimental Results

Figures 4–6 present the performance of the models in complex visual perception scenarios for object detection and keypoint localization. As shown in Figure 4, in a typical cluttered environment, all models accurately predicted both object locations and keypoints, validating the effectiveness of deep learning techniques in standard settings. In the partially occluded scenario of Figure 5, Keypoint RCNN and RTMPose missed some keypoints, while HRNet and the proposed method successfully predicted both the object and keypoints. Under low-light conditions in Figure 6, Keypoint RCNN failed to detect the object, and although RTMPose roughly identified the bounding box, it missed keypoints. HRNet accurately detected the keypoints but showed slight inaccuracies in bounding box prediction. In contrast, the proposed method delivered precise predictions for bounding boxes and keypoints across all scenarios.



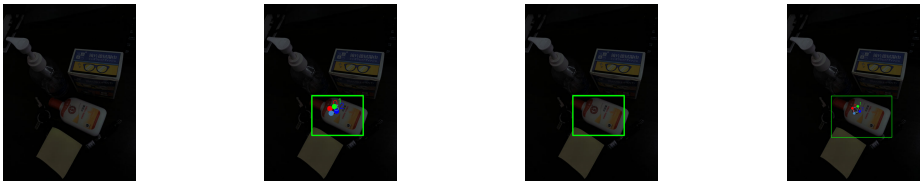
(a)keypoint RCNN (b)HRNet (c)RTMPose (d)DVSFN

Figure 4. Detection results in dense placement scenarios.



(a)keypoint RCNN (b)HRNet (c)RTMPose (d)DVSFN

Figure 5. Detection results in partial occlusion scenarios.



(a)keypoint RCNN (b)HRNet (c)RTMPose (d)DVSFN

Figure 6. Detection results in low-light scenarios.

As demonstrated by our quantitative results, in addition to utilizing the more efficient Darknet-53 backbone, we have enhanced keypoint localization through the introduction of a dynamic coordinate classifier, which improves the model's ability to localize multi-scale objects. The Gaussian positional encoding, with its smoothness and localization properties, allows for more effective capture of local spatial relationships in dense prediction tasks. Furthermore, by incorporating sparse computation, our approach accelerates inference by approximately 5% without sacrificing performance. The Adaptive Maximum Likelihood Estimation (A-MLE) component dynamically adjusts based on sample difficulty and uncertainty, thus improving the model's robustness to occlusion and low-light conditions.

5. Physical Grasping Experiment

The grasping platform consists of a UR3 collaborative robotic arm, a DH AG95 parallel gripper, a RealSense D435i sensor, and a desktop computer with the following specifications: an Intel i7 CPU, 32GB of RAM, 2TB of storage, and an NVIDIA GeForce RTX 3060 GPU. The software platform includes two main modules: feature extraction and motion control, with real-time data communication facilitated through the Robot Operating System (ROS) message-passing framework.

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, and the experimental conclusions that can be drawn.

5.1. Experiment Setting and Evaluation Criteria

Image Feature Error: The image feature error is quantified by calculating the Euclidean distance between the current keypoint coordinates and their desired locations. It is expressed as the root mean square error (RMSE) of the four keypoints, defined as

$$E_{\text{IBVS}} = \sqrt{\frac{1}{4}\sum_{i=1}^{4}\left[\left(x_i - x_i^*\right)^2 + \left(y_i - y_i^*\right)^2\right]} \tag{14}$$

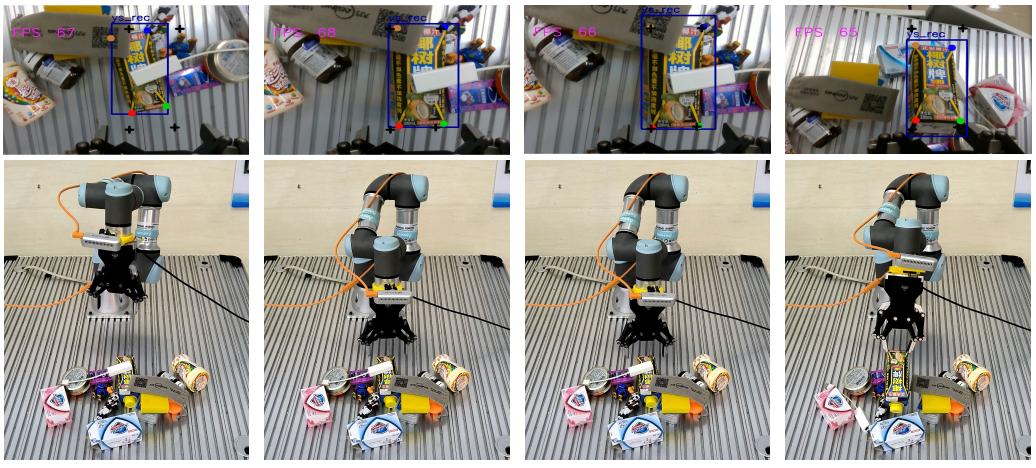
where (x_i, y_i) represent the pixel coordinates of the i-th keypoint in the current frame, and (x_i^*, y_i^*) are the expected coordinates.

Grasp Success Rate: A grasp is considered successful when the object is securely held without falling. The grasp success rate is calculated as the ratio of successful grasps to the total number of attempts.

In addition, we have also employed the grasp success rate (GSR), positioning error (PE), and grasp pose similarity (GPS) metrics, as described in [28], to provide a comprehensive evaluation of the grasping performance.

5.2. Grasping with Partial Occlusion

Grasping experiments under partial occlusion conditions were carried out using the static occlusion setup depicted in Figure 7. Various occluding objects were manually placed around the target, with some stacked directly on its surface, obscuring part of its visual features. The occlusion area was randomized, ensuring it did not exceed one-third of the target's surface. The dataset includes six object categories, each undergoing five grasping attempts. The robot's initial and grasping poses were randomized, and the visual servoing process terminated when the root mean square error of keypoint detection fell below 0.01 or when the keypoints moved outside the camera's field of view.

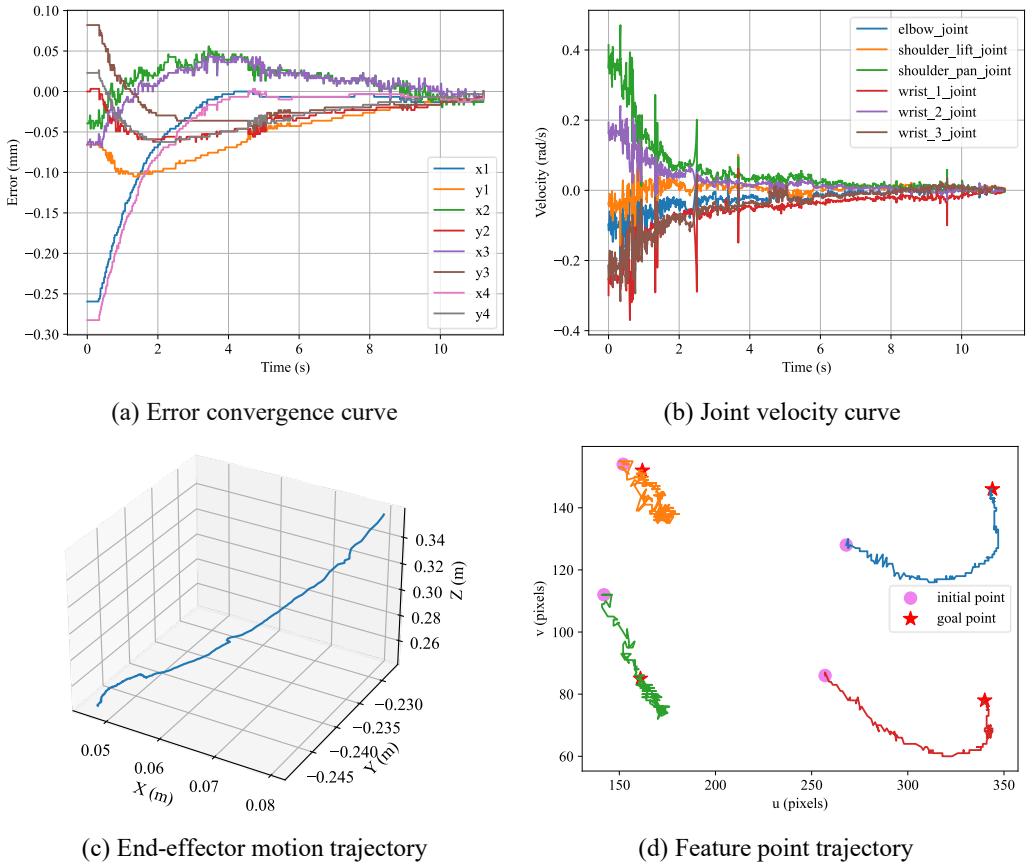


(a) (b) (c) (d)

Figure 7. Grasping experiment in partial occlusion scenario: (**a**) initial pose; (**b**) tracking pose; (**c**) grasping pose; (**d**) grasping completion.

Figure 7 illustrates the visual control process. From the external camera's viewpoint, the robot's approach to the target is visible as it gradually moves toward the object and completes the grasp. In the robot's camera view, deviations in bounding box positioning were observed during tracking, likely due to occlusion-induced errors. Despite this, keypoint features were reliably detected throughout, ensuring effective convergence of the visual servoing.

Figure 8 presents the servoing performance. Figure 8a shows that the image feature error steadily decreases after the servo task begins, successfully converging within 11 s. The fluctuations in the green and red trajectories highlight reduced stability in keypoint inference under occlusion, leading to coordinate variations. In Figure 8b, sharp velocity spikes occur at 0.5 s, 2.5 s, 3.7 s, and 9.7 s, likely due to substantial deviations in keypoint detection during servoing. The sawtooth patterns in Figure 8c,d indicate slight oscillations in the robot's motion, though it converges.



(a) Error convergence curve

(b) Joint velocity curve

(c) End-effector motion trajectory

(d) Feature point trajectory

Figure 8. Performance curves of grasping experiment in partial occlusion.

Table 2 compares the grasping performance of four visual servoing techniques under static occlusion conditions. The occlusion, which obstructed key visual features of the target, caused a significant decline in performance across most models. The IBVS method failed to successfully grasp due to its inability to detect image features effectively. The KOVIS method suffered from substantial feature loss under occlusion, resulting in a low grasp success rate of only 0.58. While the SIFT method benefited from redundancy in feature points, leading to a slight improvement in performance, the proposed method demonstrated superior robustness to occlusion, achieving a grasp success rate of 0.77.

Table 2. Grasping performance comparison in partial occlusion scenarios.

Model	GSR	SSR	PE (cm)	GPS
IBVS [29]	\	\	\	\
SIFT [30]	0.62 (19/30)	0.70 (21/30)	0.64, 3.09, 0.45	0.786
KOVIS [31]	0.58 (18/30)	0.67 (20/30)	0.72, 3.46, 0.51	0.742
Ours	**0.77 (23/30)**	**0.80 (24/30)**	**0.41, 1.98, 0.29**	**0.942**

5.3. Grasping in Low-Light Environments

In low-light conditions, images captured by the visual sensor typically display reduced contrast, color distortion, and a loss of detail in dark regions, undermining the stability of feature extraction. As a result, low-light environments are critical for assessing uncertainty in visual perception. To this end, experiments were conducted in a low-light, cluttered setting, as shown in Figure 9. Artificial lighting in the lab was turned off, and external natural light sources were partially occluded to simulate a low-illumination environment. Various interfering objects were randomly placed around the target, though the target's surface remained unobstructed. The dataset included all six object categories as candidate grasp targets, with five trials performed for each. The robot's initial and grasping poses were randomly selected, and visual servoing was terminated when the root mean square (RMS) error of the keypoint image fell below 0.01 or if the keypoints were outside the camera's field of view.

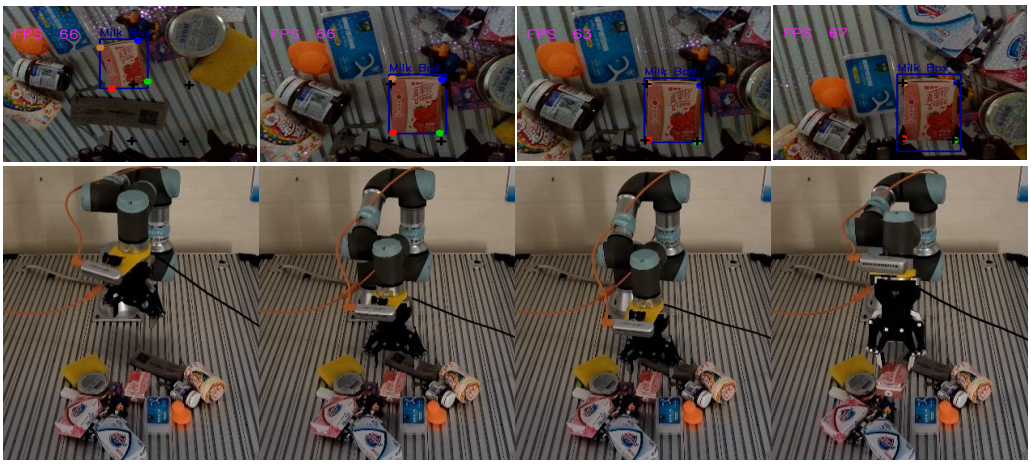


Figure 9. Grasping experiment in low-light environment: (**a**) initial pose; (**b**) tracking pose; (**c**) grasping pose; (**d**) grasping completion.

Figure 9 illustrates the performance of the proposed method during the visual servo control process. From the external camera's perspective, the robot gradually approaches the target and completes the grasping task. In the robot's camera view, the low-light environment results in diminished brightness and contrast, with noticeable noise in the upper regions of the image. Despite these challenges, both the bounding box and keypoint positions remained accurately identified throughout the process, ensuring the success of the visual servoing task.

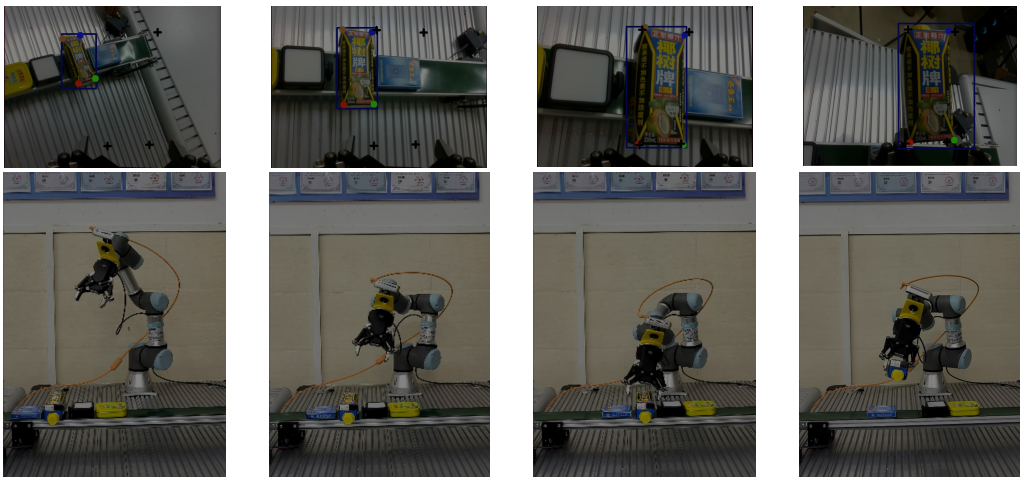
Table 3 summarizes the grasping performance of various methods in low-light conditions. In this environment, most models experienced a significant drop in grasping success. The IBVS method failed to detect the target features and thus could not perform any successful grasps. SIFT and KOVIS algorithms also struggled in low light, exhibiting frequent missed or incorrect detections, with their success rates decreasing by approximately 72% and 26%, respectively, compared to normal lighting conditions. In contrast, the proposed method maintained a grasp success rate of 0.83 under low-light conditions, demonstrating its superior adaptability and robustness.

Table 3. Grasping performance comparison in low-light environment.

Model	GSR	SSR	PE (cm)	GPS
IBVS	\	\	\	\
SIFT	0.17 (5/30)	0.23 (7/30)	0.79, 4.81, 0.92	0.726
KOVIS	0.43 (13/30)	0.50 (15/30)	0.61, 3.96, 0.64	0.782
Ours	**0.83 (25/30)**	**0.80 (24/30)**	**0.45, 2.07, 0.39**	**0.941**

5.4. Moving Object Grasping in Low-Light Conditions

In this subsection, we replicate the low-light task environment from Section 5.2, where the target object moves at a steady speed of 0.1 m/s along a conveyor belt. A successful grasping instance is shown in Figure 10. From the perspectives of the robotic arm's camera in Figure 10a,b, it is evident that the error in the image features has substantially decreased from an initial large value, suggesting that the robot is rapidly approaching the target. Figure 10c indicates that the current image features are now aligned with the expected features, confirming that the robot has reached the desired grasping position. Figure 10d depicts the robot successfully grasping the moving target. Throughout the task, the key points of the moving target were consistently and accurately detected.



(a) (b) (c) (d)

Figure 10. Moving object grasping in low-light conditions: (**a**) initial pose; (**b**) tracking pose; (**c**) grasping pose; (**d**) grasping completion.

Table 4 compares the grasping performance of moving targets in low-light environments. The IBVS, SIFT, and KOVIS methods, which rely on proportional control strategies, exhibit significant steady-state errors, limiting their grasping success rates, with none achieving successful grasping. In contrast, the proposed method shows a slight reduction in both the grasping success rate (0.73) and the servo success rate (0.83) compared to static, cluttered environments, but it still demonstrates a high level of performance. This suggests that the proposed approach maintains strong adaptability and responsiveness to moving targets in complex visual conditions.

Table 4. Moving object grasping performance comparison in low-light environment.

Model	GSR	SSR	PE (cm)	GPS
IBVS	\	\	\	\
SIFT	\	\	\	\
KOVIS	\	\	\	\
Ours	**0.73 (22/30)**	**0.83 (25/30)**	**0.52, 3.58, 0.47**	**0.943**

Although our method demonstrates strong performance in most scenarios, it is important to acknowledge certain limitations. The depth information required for image Jacobian calculation is obtained through the time-of-flight measurement method of an RGB-D camera. In specific scenarios, such as grasping tasks involving objects with special materials—encompassing transparent objects or those with highly diffuse reflective surfaces—this method may experience reduced effectiveness due to challenges in accurately capturing depth information.

The proposed grasping method was successfully tested for hardware compatibility, showing seamless adaptation to both 4-DOF and 7-DOF robots. This adaptation requires only a simple modification of the number of columns in the image Jacobian matrix. Additionally, as the method relies on 2D image features, it exhibits a high tolerance to hand–eye calibration errors, ensuring robustness across various camera setups.

6. Conclusions

This paper presents a feature network capable of detecting object-bound boxes and keypoints features, making it inherently suitable for IBVS. By optimizing the backbone network and incorporating an adaptive dynamic coordinate classifier along with sparse computation, we reduce computational complexity while enhancing accuracy. The proposed model achieves a box mAP of 0.805 and a keypoint mAP of 0.870, with an inference speed of 39 FPS. Experimental results demonstrate strong detection performance under low-light and occlusion conditions.

We propose a closed-loop grasping approach that combines DVSFN with LM-IBVS, validated through real-world experiments. The results confirm the method's robustness and versatility in challenging environments, such as occlusions and low illumination. The proposed method outperforms existing IBVS approaches in grasping tasks, offering a reliable localization solution with significant potential for real-world robotic applications.

Author Contributions: Conceptualization, J.L.; methodology, J.L.; validation, Z.Z. and R.F.; writing—original draft preparation, and review and editing, J.L. and R.F.; visualization, supervision, project administration and funding acquisition, Z.Z. and Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Guangxi Natural Science Foundation of China (2021GXNSFAA220048, 2024GXNSFAA010237) and in part by the Guangxi Science and Technology Major Program (AA24206003).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Platt, R. Grasp learning: Models, methods, and performance. *Annu. Rev. Control Robot. Auton. Syst.* **2023**, *6*, 363–389. [CrossRef]
2. Wang, S.; Sun, L.; Li, M.; Wang, P.; Zha, F.; Guo, W.; Li, Q. Learning an Image-Based Visual Servoing Controller for Object Grasping. *Int. J. Humanoid Robot.* **2024**, *21*, 2350033. [CrossRef]
3. Hutchinson, S.; Hager, G.D.; Corke, P.I. A tutorial on visual servo control. *IEEE Trans. Robot. Autom.* **1996**, *12*, 651–670. [CrossRef]
4. Billings, G.; Camilli, R.; Johnson-Roberson, M. Hybrid visual SLAM for underwater vehicle manipulator systems. *IEEE Robot. Autom. Lett.* **2022**, *7*, 6798–6805. [CrossRef]
5. Liu, H.; Li, D.; Jiang, B.; Zhou, J.; Wei, T.; Yao, X. MGBM-YOLO: A faster light-weight object detection model for robotic grasping of bolster spring based on image-based visual servoing. *J. Intell. Robot. Syst.* **2022**, *104*, 77. [CrossRef]

6. Luo, B.; Chen, H.; Quan, F.; Zhang, S.; Liu, Y. Natural feature-based visual servoing for grasping target with an aerial manipulator. *J. Bionic Eng.* **2020**, *17*, 215–228. [CrossRef]
7. Huang, H.; Zhou, H.; Sheng, M.W. Underwater vehicle visual servo and target grasp control. In Proceedings of the 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), Qingdao, China, 3–7 December 2016; pp. 1619–1624.
8. Huang, F.; Shen, C.; Mei, D.; Chen, Z. Novel IBVS System Design for Cable-Driven Hyper-Redundant Manipulator with MLESAC-Based Feature Vector Optimization. *IEEE Trans. Ind. Inform.* **2023**, *20*, 3327–3338. [CrossRef]
9. Guo, J.; Nguyen, H.T.; Liu, C.; Cheah, C.C. Convolutional neural network-based robot control for an eye-in-hand camera. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 4764–4775. [CrossRef]
10. Saxena, A.; Pandya, H.; Kumar, G.; Gaud, A.; Krishna, K.M. Exploring convolutional networks for end-to-end visual servoing. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3817–3823.
11. Bateux, Q.; Marchand, E.; Leitner, J.; Chaumette, F.; Corke, P. Training deep neural networks for visual servoing. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 3307–3314.
12. Raj, P.; Namboodiri, V.P.; Behera, L. Learning to switch CNNs with model agnostic meta learning for fine precision visual servoing. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 10210–10217.
13. Harish, Y.; Pandya, H.; Gaud, A.; Terupally, S.; Shankar, S.; Krishna, K.M. Dfvs: Deep flow guided scene agnostic image based visual servoing. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 9000–9006.
14. Adrian, N.; Do, V.T.; Pham, Q.C. DFBVS: Deep feature-based visual servo. In Proceedings of the 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), Mexico City, Mexico, 20–24 August 2022; pp. 1783–1789.
15. Lenz, I.; Lee, H.; Saxena, A. Deep learning for detecting robotic grasps. *Int. J. Robot. Res.* **2015**, *34*, 705–724. [CrossRef]
16. Depierre, A.; Dellandréa, E.; Chen, L. Jacquard: A large scale dataset for robotic grasp detection. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 3511–3516.
17. Fang, H.S.; Wang, C.; Gou, M.; Lu, C. Graspnet-1billion: A large-scale benchmark for general object grasping. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11444–11453.
18. Chen, Y.; Wu, Y.; Zhang, Z.; Miao, Z.; Zhong, H.; Zhang, H.; Wang, Y. Image-based visual servoing of unmanned aerial manipulators for tracking and grasping a moving target. *IEEE Trans. Ind. Inform.* **2022**, *19*, 8889–8899. [CrossRef]
19. Enyedy, A.; Aswale, A.; Calli, B.; Gennert, M. Stereo image-based visual servoing towards feature-based grasping. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 7325–7331.
20. Xu, L.; Guan, Y.; Jin, S.; Liu, W.; Qian, C.; Luo, P.; Ouyang, W.; Wang, X. Vipnas: Efficient video pose estimation via neural architecture search. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 16072–16081.
21. Yang, J.; Zeng, A.; Li, F.; Liu, S.; Zhang, R.; Zhang, L. Neural interactive keypoint detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 15122–15132.
22. Li, Y.; Yang, S.; Liu, P.; Zhang, S.; Wang, Y.; Wang, Z.; Yang, W.; Xia, S.T. Simcc: A simple coordinate classification perspective for human pose estimation. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Cham, Switzerland, 2022; pp. 89–106.
23. Lu, P.; Jiang, T.; Li, Y.; Li, X.; Chen, K.; Yang, W. RTMO: Towards High-Performance One-Stage Real-Time Multi-Person Pose Estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 1491–1500.
24. Horváth, D.; Erdős, G.; Istenes, Z.; Horváth, T.; Földi, S. Object detection using sim2real domain randomization for robotic applications. *IEEE Trans. Robot.* **2022**, *39*, 1225–1243. [CrossRef]
25. Yang, L.; Song, Q.; Wang, Z.; Hu, M.; Liu, C. Hier R-CNN: Instance-level human parts detection and a new benchmark. *IEEE Trans. Image Process.* **2020**, *30*, 39–54. [CrossRef] [PubMed]
26. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5693–5703.
27. Jiang, T.; Lu, P.; Zhang, L.; Ma, N.; Han, R.; Lyu, C.; Li, Y.; Chen, K. Rtmpose: Real-time multi-person pose estimation based on mmpose. *arXiv* **2023**, arXiv:2303.07399.
28. Luo, J.; Zhu, L.; Li, L.; Hong, P. Robot Visual Servoing Grasping Based on Top-Down Keypoint Detection Network. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 5000511. [CrossRef]
29. Chaumette, F.; Hutchinson, S. Visual servo control. I. Basic approaches. *IEEE Robot. Autom. Mag.* **2006**, *13*, 82–90. [CrossRef]

30. Zhang, Y.; Xu, A.; Chen, Y.; Meng, M.Q.H.; Liu, L. Multi-Scale Visual Servoing Framework for Optical Microscopy based on SIFT Matching. *IEEE Robot. Autom. Lett.* **2023**, *8*, 8144–8151. [CrossRef]
31. Puang, E.Y.; Tee, K.P.; Jing, W. Kovis: Keypoint-based visual servoing with zero-shot sim-to-real transfer for robotics manipulation. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 7527–7533.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.