IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 April 2024, accepted 2 May 2024, date of publication 6 May 2024, date of current version 14 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3397718

RESEARCH ARTICLE

Attention-Based Grasp Detection With Monocular Depth Estimation

PHAN XUAN TAN[1], (Member, IEEE), DINH-CUONG HOANG[2], ANH-NHAT NGUYEN[3], VAN-THIEP NGUYEN[3], VAN-DUC VU[3], THU-UYEN NGUYEN[3], NGOC-ANH HOANG[3], KHANH-TOAN PHAN[3], DUC-THANH TRAN[3], DUY-QUANG VU[3], PHUC-QUAN NGO[2], QUANG-TRI DUONG[2], NGOC-TRUNG HO[3], CONG-TRINH TRAN[3], VAN-HIEP DUONG[3], AND ANH-TRUONG MAI[3]

[1]College of Engineering, Shibaura Institute of Technology, Tokyo 135-8548, Japan
[2]Greenwich Vietnam, FPT University, Hanoi 10000, Vietnam
[3]IT Department, FPT University, Hanoi 10000, Vietnam

Corresponding author: Dinh-Cuong Hoang (cuonghd12@fe.edu.vn)

This work was supported by JSPS KAKENHI under Grant 24K20797.

ABSTRACT Grasp detection plays a pivotal role in robotic manipulation, allowing robots to interact with and manipulate objects in their surroundings. Traditionally, this has relied on three-dimensional (3D) point cloud data acquired from specialized depth cameras. However, the limited availability of such sensors in real-world scenarios poses a significant challenge. In many practical applications, robots operate in diverse environments where obtaining high-quality 3D point cloud data may be impractical or impossible. This paper introduces an innovative approach to grasp generation using color images, thereby eliminating the need for dedicated depth sensors. Our method capitalizes on advanced deep learning techniques for depth estimation directly from color images. Instead of relying on conventional depth sensors, our approach computes predicted point clouds based on estimated depth images derived directly from Red-Green-Blue (RGB) input data. To our knowledge, this is the first study to explore the use of predicted depth data for grasp detection, moving away from the traditional dependence on depth sensors. The novelty of this work is the development of a fusion module that seamlessly integrates features extracted from RGB images with those inferred from the predicted point clouds. Additionally, we adapt a voting mechanism from our previous work (VoteGrasp) to enhance robustness to occlusion and generate collision-free grasps. Experimental evaluations conducted on standard datasets validate the effectiveness of our approach, demonstrating its superior performance in generating grasp configurations compared to existing methods. With our proposed method, we achieved a significant 4% improvement in average precision compared to state-of-the-art grasp detection methods. Furthermore, our method demonstrates promising practical viability through real robot grasping experiments, achieving an impressive 84% success rate.

INDEX TERMS Pose estimation, robot vision systems, intelligent systems, deep learning, supervised learning, machine vision.

I. INTRODUCTION

Grasp configuration generation is a pivotal aspect of robotic manipulation, with vision-based methodologies serving as key contributors to addressing this intricate challenge [1], [2], [3]. While model-based grasp generation has been prevalent, its limitations become increasingly apparent, particularly when dealing with unknown objects [4], [5], [6]. The reliance on pre-defined 3D models and grasp databases presents significant constraints in real-world scenarios where robots encounter diverse, unmodeled objects. Conventionally, model-based grasp generation methods follow a two-step process. Firstly, a 6D object pose estimation algorithm is employed to align a Computer-Aided Design (CAD)

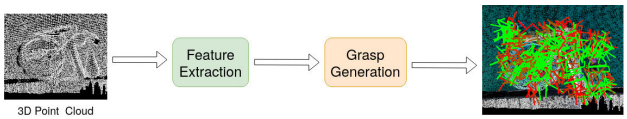
The associate editor coordinating the review of this manuscript and approving it for publication was Wai-Keung Fung.

model with measured data, providing an understanding of the object's spatial orientation. Subsequently, grasps are selected from a pre-computed database based on this alignment. However, these approaches face challenges when synthesizing grasps for unknown objects, as they presuppose the availability of a 3D model and a pre-defined grasp database [5].

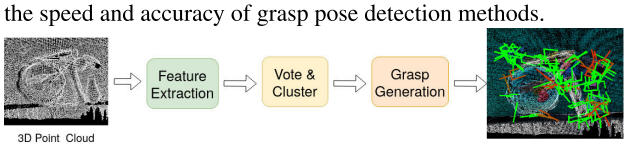
An alternative approach to grasp configuration generation, also referred to as grasp detection, involves deriving grasp configurations directly from sensor data, without presuming knowledge of the object's 3D model or relying on pre-computed grasps. This methodology is commonly referred to as grasp generation or grasp detection [2], [3], [7]. Current methods within this domain can be broadly categorized into two groups: planar grasping and six Degrees of Freedom (6)-DoF) grasping. Planar grasping employs a straightforward yet effective representation, defining grasps in terms of oriented bounding boxes [8], [9]. This low degree of freedom (DoF) representation simplifies the task into a detection problem but may limit performance in more complex 3D manipulation tasks. On the other hand, 6-DoF grasping provides greater dexterity and is more suitable for handling intricate scenarios [2], [10]. However, the accurate generation of 6-DoF grasps often requires geometric information, leading many existing methods to depend on 3D point cloud data.

While grasp generation methods utilizing point clouds have shown progress, they still face significant challenges in cluttered scenes due to lack of visual cues [5], [7], [11]. Without RGB information, these methods may struggle to distinguish between objects with similar depth profiles but different visual appearances. This limitation can result in difficulty differentiating between objects of interest and background clutter, leading to inaccurate grasp point detection. RGB information can provide additional contextual information about object texture, color, and shape, which can improve the robustness and reliability of grasp detection algorithms. To address the limitations associated with using either depth or RGB data alone, many researchers have turned to methods that leverage both modalities simultaneously [12], [13], [14], [15]. By combining depth and RGB information, these approaches aim to exploit the complementary strengths of each modality while mitigating their respective weaknesses. This integration of multiple modalities has led to significant advancements in various computer vision tasks [16], [17], [18], [19], including grasp detection [20], [21]. However, existing approaches require complex multi-stage processing, which can be time-consuming and computationally intensive. In addition, the RGBD methods typically rely on specialized hardware like depth cameras or stereo camera setups [16], [22]. This dependency on specific hardware can limit the accessibility and scalability of these methods. In contrast, acquiring only RGB images is a more cost-effective and straightforward approach compared to obtaining RGBD data. The potential of leveraging RGB images for grasp detection offers several advantages. Firstly, RGB cameras are more commonly available and less expensive than depth sensors, making them more accessible for various applications. Secondly, deep learning techniques have shown effectiveness in depth estimation from RGB data as well as understanding and interpreting RGB images [23], [24], which can be leveraged for grasp detection tasks. Exploring grasp detection from RGB images and predicted depth data opens up new avenues for research and development in robotics and automation. However, grasp detection from RGB images with depth estimation remains largely unexplored. Current RGBD fusion methods [13], [14], designed for integrating data from depth sensors, may not be directly applicable to predicted depth maps generated from RGB images using depth estimation algorithms. Predicted depth maps may suffer from inaccuracies or noise introduced during the depth estimation process, leading to discrepancies with true depth values. Additionally, these maps may exhibit different noise characteristics compared to depth data obtained from sensors. For example, depth estimation algorithms may introduce specific types of noise or artifacts not present in sensor data, posing challenges for existing RGBD fusion methods. Addressing these challenges requires the development of novel fusion techniques specifically tailored to handle the characteristics and noise patterns associated with predicted depth maps, thereby enabling more effective grasp detection from RGB images with depth estimation.

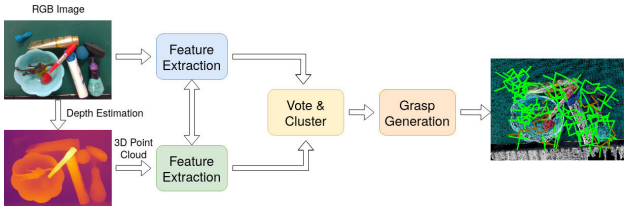
This study introduces a deep learning framework designed for model-free 6-DoF grasping, exclusively relying on an RGB image for accurate grasp estimation, building upon our previous work VoteGrasp [2]. While VoteGrasp achieved promising results with its voting module, it suffered from limitations due to the absence of appearance information. In contrast to VoteGrasp, which relies solely on 3D point cloud data, our proposed method integrates both color and depth images to extract discriminative features and generate collision-free grasps. Figure 1 illustrates the distinctions between the proposed approach, VoteGrasp, and conventional methods. The key components of our developed system include depth estimation, attention-based adaptive fusion utilizing visual-guided 3D geometric feature learning (VGG) and geometric-guided visual feature learning (GGV), and voting-based grasp generation. While the voting module is adapted from VoteGrasp, the other components are novel. To extract essential geometric information for prediction, we leverage recent advancements in monocular depth estimation to generate 3D point clouds. Notably, this represents the first deep learning network utilizing 3D point clouds derived from estimated depth maps for grasp generation. Given an RGB image and a predicted 3D point cloud, our proposed method incorporates adaptive fusion modules to extract discriminative features. We facilitate bidirectional information flow through VGG and GGV modules, enabling mutual utilization of local and global information and enhancing the representation learning process for both branches. We evaluate the proposed method on both a



(a) Traditional methods for grasp pose detection often employ uniform sampling of points across a scene, treating all points equally [7], [11], [25]. However, this approach overlooks the critical consideration of where to grasp, which significantly compromises both the speed and accuracy of grasp pose detection methods.



(b) VoteGrasp [2] employs a voting mechanism to identify graspable areas, focusing the system's efforts on specific points or regions of an object suitable for grasping. However, using only 3D point cloud as input, performance is often inadequate due to the lack of visual cues.



(c) The proposed method in this research utilizes both color and depth images as input to extract discriminative features and generate collision-free grasps through voting. Instead of relying solely on depth sensors, we explore the use of predicted depth images from an off-the-shelf depth estimation framework.

FIGURE 1. Comparison between the proposed approach and conventional methods, as well as our previous work, VoteGrasp [2].

standard dataset and in a real robot grasping application. The results showcase that, even when utilizing only RGB images with estimated depth maps, our approach outperforms state-of-the-art methods that rely on depth images from sensors.

The key contributions of this study include:

- **Innovative Grasp Detection Network**: We present a deep learning network designed specifically for grasp detection, leveraging estimated depth maps alongside RGB images, thus eliminating the reliance on measured depth images.
- **Attention-Based Adaptive Fusion**: We introduce an attention-based adaptive fusion technique that incorporates visual-guided 3D geometric feature learning (VGG) and geometric-guided visual feature learning (GVV), enhancing the robustness and accuracy of our model.
- **Voting-Based Framework**: We integrate our adaptive fusion module with the voting-based grasp generation module from VoteGrasp to enhance resilience to noise and occlusion. This framework effectively improves the system's robustness in cluttered scenes with multiple objects.
- **Performance**: Through extensive experimentation on a publicly available dataset [3], our proposed approach consistently outperforms state-of-the-art methods. Furthermore, we validate its practical applicability by integrating the system with real robot platforms and conducting successful grasping experiments.

TABLE 1. Vision-based grasp detection methods.

Method	RGB	Depth	6-DoF	End-to-End
Lenz et al. [9]	yes	yes	no	no
Readmon [26]	yes	no	no	yes
GPD [7]	no	yes	yes	no
PointNetGPD [11]	no	yes	yes	no
Lou et al. [25]	no	yes	yes	no
Kokic et al. [27]	no	yes	yes	no
Schmidt et al. [28]	no	yes	yes	yes
Yang et al. [29]	no	yes	yes	yes
GraspNet [3]	no	yes	yes	yes
PointNet++ [30]	no	yes	yes	yes
VoteGrasp [2]	no	yes	yes	yes
Ours	yes	yes	yes	yes

The remainder of this article is organized as follows: Section II, Related Work, provides an overview of Learning-based Grasp Generation (II.A), Monocular Depth Estimation (II.B), and RGBD Fusion (II.C). In Section III, Methodology, we introduce the proposed framework. This includes Depth Estimation (III.A) and details our innovations: the Attention-based Adaptive Fusion Network (III.B) incorporating Visual-Guided 3D Geometric Feature Learning and Geometric-Guided Visual Feature Learning, as well as the Voting-based Grasp Generation (III.C) approach. Section IV, Evaluation, includes Dataset (IV.A) and the Implementation Details subsection (IV.B), discussing the technical specifics of our methods. The Evaluation on GraspNet-1Billion (IV.C) subsection presents the results and analysis based on the GraspNet-1Billion dataset. Finally, the Robotic Grasping Experiment (IV.D) subsection provides insights derived from real-world experiments in robotic grasping. Lastly, Section V, Conclusions, provides a summary of the key contributions and findings presented in this article. It also outlines potential avenues for future research, paving the way for further advancements.

II. RELATED WORK

In this section, we review relevant works, specifically focusing on existing vision-based grasp detection methods, monocular depth estimation, and RGBD fusion.

A. VISION-BASED GRASP DETECTION

Vision-based grasp detection for robot manipulation refers to the use of visual information, typically obtained from cameras or other imaging devices, to identify suitable grasp poses on objects [7], [9], [10]. The goal is to enable a robot to autonomously plan and execute grasping actions with precision. Table 1 provides a comparative overview of various vision-based grasp detection methods, highlighting their utilization of RGB and depth information, ability to estimate 6-DoF grasp poses, and whether they offer

end-to-end learning. Earlier approaches [31], [32] assumed complete 2D or 3D object knowledge or simplified objects as primitive shapes, facing limitations in obtaining accurate 3D models. Learning-based methods have emerged, leveraging large-scale data and automated feature extraction. Some initially focused on 4-DoF grasp poses on the camera plane, referred to as top-down grasping [8], [9], [26]. However, this approach restricts degrees of freedom and may miss crucial grasp poses, especially those along object edges. In contrast, 6-DoF grasp poses offer increased flexibility and complexity, allowing grasping from various directions [7], [10], [11], [25]. They necessitate six parameters to define location and rotation, with the potential inclusion of additional degrees of freedom, such as gripper width or height. Learning-based grasp generation can be categorized into two primary algorithmic methodologies for grasp synthesis: grasp pose sampling and direct regression of grasp pose. Sampling-based approaches, like GPD [7] and PointNetGPD [11], evaluate individual grasp samples. However, despite dense sampling, they struggle in regions like the rims of objects where surface normals estimation is unreliable. Some methods, such as Redmon and Angelova [25], sample wrist angles independently, while others, like Kokic et al. [27], sample grasp, roll angles, and offset distances. These approaches often trade computation time for generated grasp poses, resulting in limited poses per scene and a focus on local object features. Direct regression methods, exemplified by Schmidt et al. [28] and Yang et al. [29], predict grasp poses or transformation matrices directly from visual data, processing information holistically. Yet, approaches like GraspNet [3] and PointNet++ [30], utilizing entire scene point clouds, lack consideration for inter-object relationships, limiting performance in cluttered scenes and under occlusion. To overcome these limitations, our previous work [2] leveraged a voting mechanism and contextual information to directly generate grasp configurations from 3D point clouds, addressing challenges in occlusion common in manipulation. The proposed method presented in this study builds upon our prior research [2]. However, instead of exclusively relying on 3D data from depth sensors, we investigate the utilization of both color and depth images for grasp detection. Especially, we incorporate depth images estimated from a monocular depth estimation framework, eliminating the need for depth sensors while enhancing the robustness and versatility of our approach.

B. MONOCULAR DEPTH ESTIMATION

Monocular Depth Estimation (MDE) is a crucial computer vision task that involves predicting the depth information of a scene using a single 2D image captured by a monocular camera. Accurate depth information is paramount for comprehending the three-dimensional structure of a scene. The origins of monocular depth estimation can be traced back to pioneering work by Saxena et al. [33], [34], which utilized hand-engineered features and Markov Random Fields (MRF). The landscape of depth estimation underwent a revolutionary transformation with the advent of deep learning, notably led by Eigen et al. [35]. However, challenges arise in learned depth regression during the decoder phase, as fine details may be lost in successive convolution layers of neural networks. Addressing this issue, [36] introduced multi-scale networks to predict depth at various resolutions, while Laina et al. [37] enhanced a ResNet architecture with improved up-sampling blocks to mitigate information loss. Xu et al. [38] combined deep learning with conditional random fields (CRF) for feature fusion at different scales. Another research direction explored multitask learning, incorporating simultaneous predictions of semantic labels [39], depth edges, and normals [40], [41], [42] to refine depth predictions. Kendall et al. [43] investigated the impact of uncertainty estimation on scene understanding, while Yin et al. [44] used surface geometry to estimate 3D point clouds from predicted depth maps. Recent works, such as that by Bhat et al., propose a classification-based formulation for distance prediction. Chen et al. [45] integrated attention blocks into the decoder, and Transformer-based architectures gained traction [46], [47]. The estimation of depth is a pivotal component in understanding geometric relations within a scene. This understanding contributes to richer representations of objects and their environment, leading to improvements in existing recognition tasks and enabling diverse applications, including 3D modeling, physics and support models, robotics, and reasoning about occlusions. In this study, we explore how monocular depth estimation can enhance the performance of grasp detection for robot manipulation.

C. RGBD FUSION

In recent years, there has been significant attention given to RGBD feature fusion, particularly in the domains of semantic segmentation [50], [51], [52], [53] and autonomous driving [54], [55], [56], [57], [58]. Chen et al. [50] introduced a unified cross-modality guided encoder for recalibrating RGB feature responses and distilling depth information across multiple stages. The innovative separation-and-aggregation gating operation jointly filters and recalibrates both representations before cross-modality aggregation. The fusion module serves the dual purpose of propagating and fusing information between modalities while preserving their specificity throughout the long-term propagation process. In [51], depth features are fused into the RGB encoder at each of the five resolution stages. Leveraging a Squeeze and Excitation (SE) module, features from both modalities are reweighted and then summed element-wise. This channel attention mechanism enables the model to learn which features to focus on and which to suppress based on the input, leading to notable improvements in segmentation. Wang et al. [52] proposed TokenFusion, a multimodal token fusion method tailored for transformer-based vision tasks. TokenFusion dynamically identifies uninformative tokens

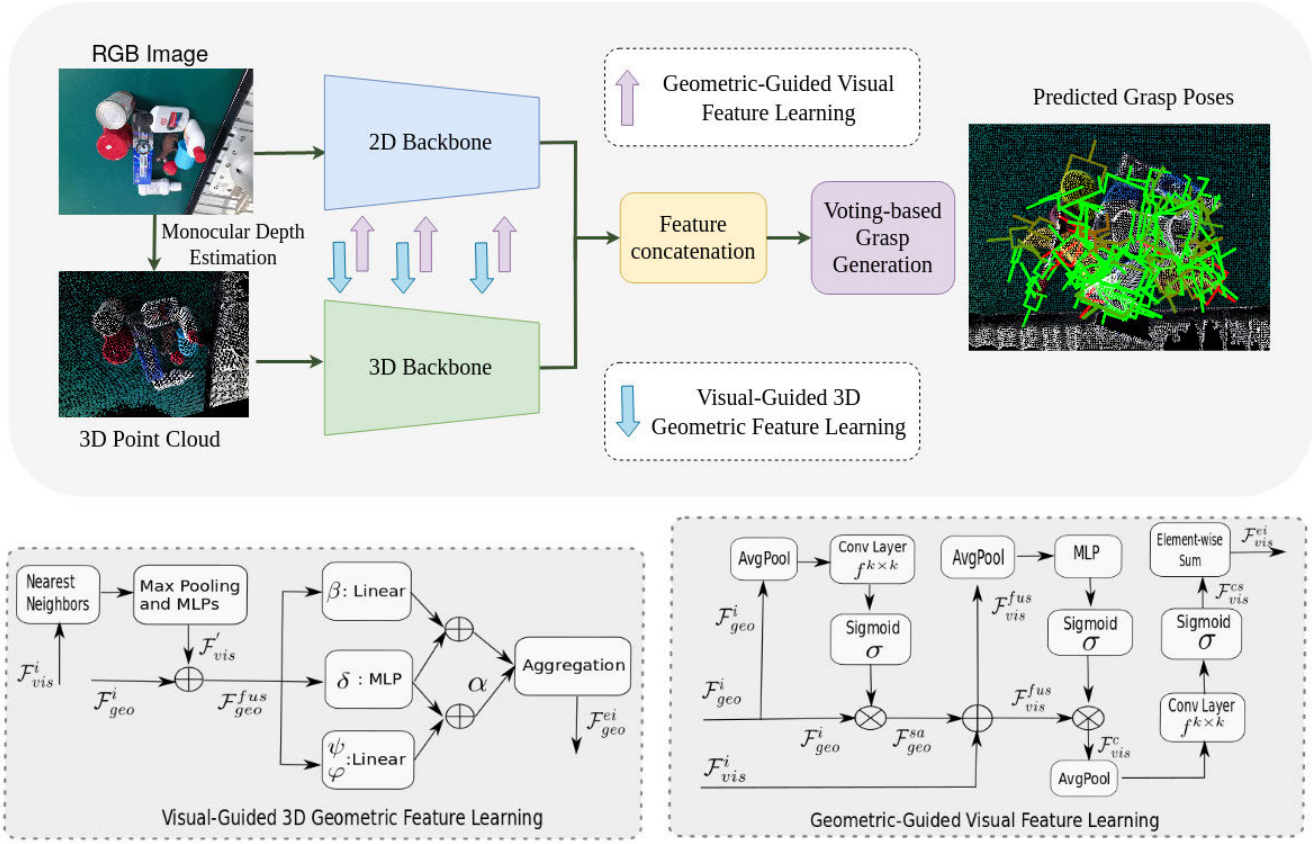


FIGURE 2. Network Architecture Overview: The diagram highlights the essential components of our network, encompassing monocular depth estimation, a 2D backbone (ResNet [48]), a 3D backbone (PointNet++ [49]), geometric-guided visual feature learning (VGG), visual-guided 3D geometric feature learning (GGV), and voting-based grasp generation [2].

and substitutes them with projected and aggregated inter-modal features. Residual positional alignment is incorporated to explicitly utilize inter-modal alignments post-fusion. TokenFusion's design allows the transformer to learn correlations among multimodal features while maintaining the largely intact structure of the single-modal transformer architecture. Addressing the limitation of fusion quality due to the sparsity of Light Detection and Ranging (LiDAR) points, Bai et al. [54] introduced a LiDAR-Camera fusion module. Instead of fetching a limited number of image features based on the hard association between LiDAR points and image pixels, this module retains all image features in a memory bank. The cross-attention mechanism in the transformer decoder then performs sparse-to-dense and adaptive feature fusion. Chen et al. [55] developed a cross-domain DeformCAFA module to tackle the computational cost issue introduced by global-wise attention networks. Similarly, [56], [57], [58] proposed camera-LiDAR fusion architectures to bridge the gap between feature representations of cameras and 3D point cloud data. Building on the motivation from these works [50], [51], we introduce a bi-directional multi-step fusion network with an attention mechanism to enhance RGB and geometry feature representation.

III. METHODOLOGY

Let $\mathcal{E}$ denote the environment, encompassing the robot and objects, and $s(\mathcal{E}, \mathbf{I}, \mathcal{G}, G)$ represent a binary variable indicating grasp success or failure. Here, G represents the Grasp Pose defined by a tuple $G = (x, y, z, rx, ry, rz, w)$, where x, y, and z denote the translation of the gripper, while rx, ry, and rz denote the rotation, and w denotes the width accordingly. $\mathbf{I} = (\mathbf{I}_v, \mathbf{I}_d)$ stands for the RGBD Image. Here, $\mathbf{I}_v = \mathbb{R}^{3\times H\times W}$ denotes the RGB image, and $\mathbf{I}_d = \mathbb{R}^{H\times W}$ denotes the depth map. For simplicity, we consider only the most common parallel-jaw gripper, with its configuration $\mathcal{G}$ defined by a tuple: $\mathcal{G} = (h, l, w_{max})$, where h, l, and w_{max} represent the height, length, and maximum width of the gripper, respectively.

Given an RGBD image $\mathbf{I}$ and a gripper configuration $\mathcal{G}$, our objective is to determine a set of grasp poses $G = \{G_1, G_2, \ldots, G_m\}$ that maximizes the grasp success rate, with m being a fixed parameter. This entails our algorithm predicting a diverse array of grasp poses to adequately cover the scene, providing multiple candidates for grasp execution. It's important to note that instead of directly capturing depth data from a sensor, we assume the availability of predicted depth maps from off-the-shelf depth estimation frameworks. To enhance the reliability of the predicted depth

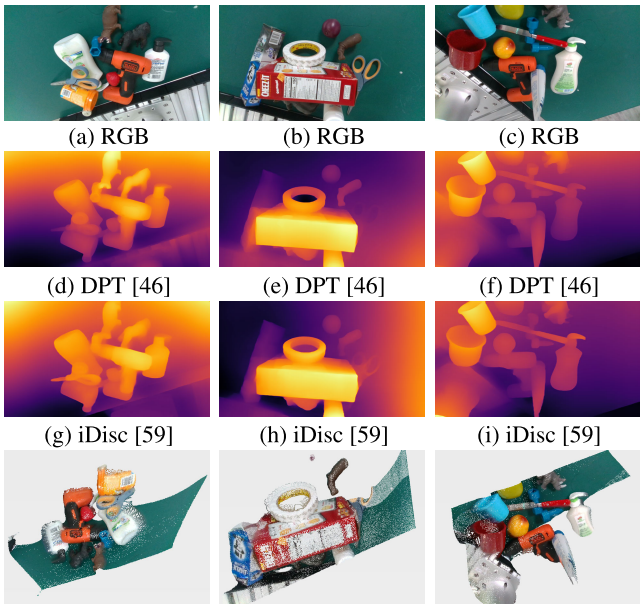


FIGURE 3. Examples of input RGB images, depth maps predicted by DPT [46] and iDisc [59], and 3D point cloud extracted from predicted depth maps.

data, we perform depth refinement before feeding it into our network.

Figure 2 provides an overview of our proposed method. The key components include depth estimation, attention-based adaptive fusion incorporating visual-guided 3D geometric feature learning (VGG) and geometric-guided visual feature learning (GVV), and voting-based grasp generation. The synergy among these components enhances feature discriminability and robustness, resulting in more accurate and efficient grasp pose generation. In the subsequent sections, we provide a comprehensive explanation of each component.

A. DEPTH ESTIMATION

Monocular depth estimation, the task of predicting the depth of a scene from a single 2D image, has seen significant advancements in recent years [23]. However, many existing methods are designed with a focus on large outdoor scenes [24], [46], [59], [60], making them less suitable for smaller objects intended for manipulation. Inspired by Kim et al.'s work [61], where they introduced a mask-guided depth refinement method using generic masks, we recognize the importance of refining depth predictions to attain higher accuracy in robotic grasping tasks. However, their method's reliance on high-quality masks for refinement introduces limitations, as the refinement performance is inherently constrained by the quality of the mask. Moreover, the integration of their method into our system presents complexities that may compromise real-world deployment due to increased computational overhead. To address these challenges, we introduce an uncertainty-based depth refinement approach. This method is simpler yet effective, focusing on enhancing depth information for robotic grasping tasks without compromising computational efficiency. We leverage two distinct depth estimation networks, DPT [46] and iDisc [59], to generate individual depth images denoted as D_1 and D_2, respectively. By calculating the disparity between these images, we can identify regions with significant differences beyond a predefined threshold τ as uncertain areas. Our approach involves excluding these uncertain regions from the depth images and replacing the depth values within other areas with their mean values. Specifically, we first compute the disparity between the two depth images:

$$\delta(x, y) = |D_1(x, y) - D_2(x, y)| \tag{1}$$

Subsequently, we compute the mean depth values:

$$\overline{D}(x, y) = \frac{D_1(x, y) + D_2(x, y)}{2} \tag{2}$$

Afterwards, we replace depth values in certain regions with the mean depth, ensuring a more robust and accurate depth image:

$$\mathbf{I}_d(x, y) = \begin{cases} \overline{D}(x, y) & if\ \delta(x, y) < \tau \\ NaN & if\ otherwise \end{cases} \tag{3}$$

Through this refinement process, we aim to enhance depth information specifically for scenes in robot manipulation application, resulting in an improved and more accurate depth image denoted as $\mathbf{I}_d$. Examples of estimated depth maps and 3D point clouds extracted from $\mathbf{I}_d$ are showcased in Figure 3.

B. ATTENTION-BASED ADAPTIVE FUSION NETWORK

Given an RGB image $\mathbf{I_v}$ and an estimated depth map $\mathbf{I_d}$, our initial step involves transforming the depth image $\mathbf{I_d}$ into a point cloud $\mathbf{P}$ using the camera intrinsic matrix. Subsequently, we utilize ResNet34 [48] for extracting visual features $\mathcal{F}_{vis}$ from the RGB image and PointNet++ [49] for obtaining geometric features $\mathcal{F}_{geo}$ from the point cloud $\mathbf{P}$. These networks enable bidirectional information flow through Visual-Guided Geometric Feature Learning (VGG) and Geometric-Guided Visual Feature Learning (GGV) modules, facilitating mutual utilization of local and global information. This enhances the representation learning process for both branches.

1) VISUAL-GUIDED 3D GEOMETRIC FEATURE LEARNING

To integrate visual information from $\mathcal{F}^i_{vis}$ into geometric features $\mathcal{F}^i_{geo}$ in the i-th stage, we introduce a novel Visual-Guided Geometric Feature Learning (VGG) module. Rather than globally compressing the RGB feature map and potentially losing intricate details, we utilize the aligned RGBD image. Each pixel's depth contributes to deriving its corresponding 3D point, establishing an XYZ map aligned with the RGB map. For every geometric feature paired with its 3D point coordinate, we retrieve visual features from $\mathcal{F}_{vis}$ by projecting its neighborhood, with a radius r_1, onto the image. Subsequently, we sample the k_1 nearest neighbor

pixels within this region, gathering their visual features. In cases where fewer than k_1 pixels exist in the corresponding region, null features are padded. These collected visual features are integrated using max pooling and processed through Multi-Layer Perceptrons (MLPs) to match their channel size with the point cloud feature. This stage produces modified visual features $\mathcal{F}'_{vis}$. Subsequently, we concatenate the integrated visual features $\mathcal{F}'_{vis}$ with the geometric features $\mathcal{F}^i_{geo}$ and apply a shared MLP to obtain the fused geometric feature $\mathcal{F}^{fus}_{geo}$. Consequently, the network enriches N 3D points with high-dimensional features, denoted as $\mathcal{P} = \{p_i\}^N_{i=1}$ and $\mathcal{F}^{fus}_{geo} = \{f_i\}^N_{i=1}$, where $p_i = [x_i; f_i]$. Here, $x_i \in \mathbb{R}^3$ signifies the point's location in 3D space, and f_i represents the associated feature vector. The enriched points $\{p_i\}^N_{i=1}$, now imbued with the fused features, are then inputted into our self-attention module to enhance the features $\mathcal{F}^{ei}_{geo}$. In accordance with [62], [63], the self-attention module is defined as follows:

$$y_i = \sum_{p_j \in \mathcal{P}(i)} (\alpha(\gamma(p_i, p_j) + \delta) \odot \beta(p_j)) \tag{4}$$

$\mathcal{P}(i) \subseteq \mathcal{P}$ refers to a set of points in the local neighborhood of p_i. α, γ, δ, and β signify a mapping function, a relation function, a position encoding function, and pointwise feature transformation, respectively. The relation function γ uses subtraction to output a vector representing the features of p_i and p_j:

$$\gamma(p_i, p_j) = \varphi(p_i) - \psi(p_j) \tag{5}$$

Here, φ and ψ represent trainable transformations using multilayer perceptrons (MLPs). The mapping function α is an MLP with two linear layers and one ReLU nonlinearity, allowing the module to compute attention weights spatially and across channels while maintaining computational efficiency. To adapt to local data structures, we introduce spatial context using a trainable and parameterized position encoding function δ:

$$\delta = \phi(x_i - x_j) \tag{6}$$

x_i and x_j denote the 3D point coordinates for points i and j, respectively. The encoding function ϕ is an MLP with two linear layers and one ReLU nonlinearity.

2) GEOMETRIC-GUIDED VISUAL FEATURE LEARNING

The Geometric-Guided Visual Feature Learning (GGV) module provides an alternative approach to integrating geometric information from $\mathcal{F}^i_{geo}$ into visual features $\mathcal{F}^i_{vis}$ during the i-th stage. Rather than naively concatenating global point features, this module densely fuses features by identifying k_2 nearest points for each pixel from the point cloud, collecting corresponding point features, and integrating them via max pooling to produce $\mathcal{F}'_{geo}$. These features are then passed through a spatial attention block M_{sa1} [64]. This mechanism is designed to discern informative regions, eliminating redundant geometric-guided features that may arise from noise or irrelevant areas, thereby facilitating a more effective integration with the visual features $\mathcal{F}^i_{vis}$. The block utilizes average-pooling to highlight informative regions, resulting in $\mathcal{F}^{avg}_{geo} \in \mathbb{R}^{W \times H}$. Subsequently, $\mathcal{F}^{avg}_{vis}$ undergoes a $k \times k$ filter convolution and normalization via the sigmoid function. The output, denoted as $M_{sa1}(\mathcal{F}^i_{geo})$, is then element-wise multiplied with the original geometric features, $\mathcal{F}^i_{geo}$, to acquire the initial enhanced geometric-guided features, $\mathcal{F}^{sa}_{geo}$. The summarized attention process is illustrated as:

$$M_{sa1}(\mathcal{F}^i_{geo}) = \sigma(f^{k\times k}(AvgPool(\mathcal{F}^i_{geo})) \tag{7}$$

$$\mathcal{F}^{sa}_{geo} = M_{sa1}(\mathcal{F}^i_{geo}) \otimes \mathcal{F}^i_{geo} \tag{8}$$

Here, $\otimes$ denotes element-wise multiplication, σ represents the sigmoid function, and $f^{k\times k}$ denotes a convolution operation utilizing a $k \times k$ filter. We empirically chose $k = 7$ following the setting in [64]. Subsequently, $\mathcal{F}^{sa}_{geo}$ is integrated with the visual features $\mathcal{F}^i_{vis}$ through element-wise summation to produce the fused features $\mathcal{F}^{fus}_{vis}$:

$$\mathcal{F}^{fus}_{vis} = \mathcal{F}^i_{vis} \oplus \mathcal{F}^{sa}_{geo} \tag{9}$$

where $\oplus$ signifies element-wise summation. To further refine the fused features $\mathcal{F}^{fus}_{vis}$, a channel attention block M_{ca} [65] is introduced. This block utilizes global average pooling to reduce each feature map within $\mathcal{F}^{fus}_{vis}$ to a single pixel, generating a 1D vector of length C. The vector undergoes an MLP network with a hidden layer and sigmoid activation, followed by element-wise multiplication with $\mathcal{F}^{fus}_{vis}$. This process recalibrates the feature responses, accentuating important channels while suppressing less relevant ones. The output of M_{ca}, denoted as $\mathcal{F}^c_{vis}$, can be summarized as:

$$M_{ca}(\mathcal{F}^{fus}_{vis}) = \sigma(MLP(AvgPool(\mathcal{F}^{fus}_{vis})) \tag{10}$$

$$\mathcal{F}^c_{vis} = M_{ca}(\mathcal{F}^{fus}_{vis}) \otimes \mathcal{F}^{fus}_{vis} \tag{11}$$

Moreover, $\mathcal{F}^c_{vis}$ undergoes re-weighting by another spatial attention block, M_{sa2}, with components akin to M_{sa1}, producing $\mathcal{F}^{cs}_{vis}$. Finally, $\mathcal{F}^{cs}_{vis}$ is integrated with the visual features $\mathcal{F}_{vis}$ through element-wise summation, yielding the enhanced feature representation $\mathcal{F}^{ei}_{vis}$.

3) FUSION

Following bidirectional fusion in both VGG and GGV modules, distinct features are extracted by the visual and geometric branches. To generate reliable correspondences and obtain more distinctive features, a simple undirected fusion is performed in the final stage. By projecting each point to the image plane with the camera intrinsic matrix, correspondences between visual and geometry features are established. These pairs are concatenated to form the extracted dense fused feature $\mathcal{F}$, subsequently utilized in the voting-based grasp generation module in the subsequent step.

C. VOTING-BASED GRASP GENERATION

Given the extracted dense fused feature $\mathcal{F} = \{f_i\}$, we predict grasp poses using the voting-based grasp generation module

in our previous work [10]. Each grasp comprises a center point $p \in \mathbb{R}^3$, a gripper orientation $R \in SO(3)$, a gripper width $w \in \mathbb{R}$, and a grasp score $q \in [0, 1]$. We generate M seeds $\{s_i\}_{i=1}^{M}$, where each seed $s_i = [x_i, f_i^s]$ holds the 3D spatial location $x_i \in \mathbb{R}^3$ and the corresponding feature vector $f_i^s \in \mathbb{R}^F$. Processing these seeds through an MLP computes J votes $\{\{v_{ij} = [y_{ij}; f_{ij}^v] \in \mathbb{R}^{3+F}\}_{i=1}^{M}\}_{j=1}^{J}$, leveraging fully connected layers, ReLU activation, and batch normalization. Each vote v_{ij} comprises a 3D point y_{ij} close to a grasp center in Euclidean space and a F-dimensional feature vector f_{ij}^v. Clustering the votes via uniform sampling and Euclidean distance identifies K votes $\{v_k\}_{k=1}^{K}$. Using iterative farthest point sampling (FPS) based on $\{y_i\}$, K clusters form from the sampled votes, employing a ball query to gather votes within a set radius of the query vote v_k.

To achieve collision-free grasps in complex environments, comprehending object relationships and contextual cues within features is essential. Our VoteNet integrates a contextual module inspired by self-attention models. It utilizes an MLP and max-pooling to process cluster votes, aggregating into $f_k^c \in \mathbb{R}^{F'}$. These vectors compile into a map $f^c = [f_1^c; f_2^c; \ldots; f_K^c] \in \mathbb{R}^{K \times F'}$, fostering inter-cluster feature communication, significantly enhancing grasp detection performance.

Following the computation of the contextual feature map, our model employs an MLP network to detect a ranked list of grasps $G = (p, R, w, q)$. The prediction layer includes $5 + V + 2A$ channels: 3 for grasp center regression values, 1 for gripper width regression value, 1 for grasp confidence regression value, V for viewpoint scores, and A each for angle scores and angle residual regression values for in-plane rotation. Here, V and A represent the numbers of sampled viewpoints and in-plane rotations, respectively.

Loss Function: The learning of modules is supervised jointly using a multi-task loss:

$$L = \lambda_1 L_{vote} + \lambda_2 L_{grasp} \tag{12}$$

The voting loss L_{vote} is a regression loss formulated as:

$$L_{vote} = \frac{1}{M_s} \sum_i \|y_i - c_i^g\|_H \cdot \mathbb{1}(x_i) \tag{13}$$

Here, M_s represents the total number of seed points on the object surface, c_i^g is the closest ground truth grasp center, $\|\cdot\|_H$ denotes the Huber norm, and $\mathbb{1}(\cdot)$ is a binary function determining whether a seed point s_i belongs to an object.

The grasp loss function L_{grasp} is defined as:

$$L_{grasp} = L_{center} + \alpha L_{rot} + \beta L_{width} + \gamma L_{score} \tag{14}$$

The L_{grasp} comprises losses for grasp center regression (L_{center}), rotation (L_{rot}), gripper width regression (L_{width}), and grasp confidence score regression (L_{score}). The grasp center loss includes viewpoint classification loss ($L_{viewpoint}$) and in-plane rotation loss ($L_{in-plane}$), which consists of classification ($L_{angle-cls}$) and regression ($L_{angle-reg}$) losses. Regression losses employ $L1$-smooth loss, while classification losses use standard cross-entropy loss [10].

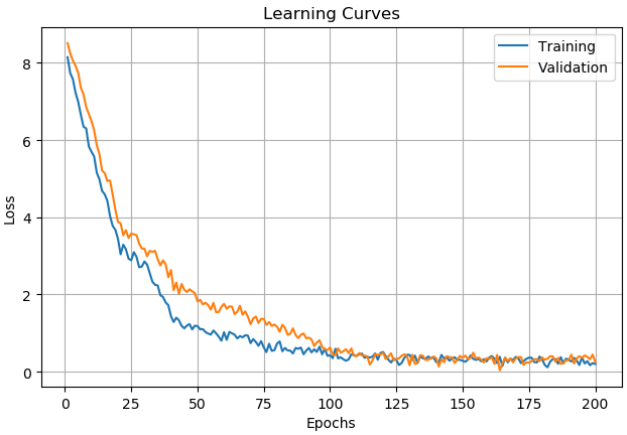


FIGURE 4. Learning curves of our model trainning on GraspNet-1Billion dataset [3].

With the loss function L in Eq. 12, our network training becomes independent of object category labels. This unique feature empowers the trained model to generate grasp configurations seamlessly, even in the absence of object category information during inference, thus enhancing its generalization to novel objects. However, real-world scenarios often involve specific target objects-items that robots are anticipated to manipulate in predefined ways or tasks. Such target objects may include items on a production line requiring sorting or assembly, or household objects needing precise picking and relocation. To effectively handle this challenge, we propose a modification that incorporates an additional loss function tailored for grasping target objects, defined as follows:

$$L = \lambda_1 L_{vote} + \lambda_2 L_{grasp} + \lambda_3 L_{sem} \tag{15}$$

The semantic classification loss, L_{sem}, utilizes cross-entropy for NC classes. When extended to target objects, this adaptation allows the model to generate grasp configurations for specific objects without the necessity of explicit object detection or pose estimation. Here, NC corresponds to the number of classes for known objects. The semantic classification loss, expressed through cross-entropy, is given by:

$$L_{sem} = \sum_{i=1}^{NC} q_i log(\hat{q}_i) \tag{16}$$

where q_i represents the true probability of the grasp belonging to object class i (ground truth), and $\hat{q}_i$ is the predicted probability for the same class.

IV. EVALUATION

To assess the efficacy of our vision-based grasp detection system, we conduct evaluations on a publicly available dataset [3]. Additionally, we integrate the system with real robot platforms and perform grasping experiments to validate its performance in practical scenarios. We chose the GraspNet

TABLE 2. Layer parameters of PointNet++ [49] based feature learning network.

layer name	input layer	layer params
SA1	point cloud	(2048,0.025,[64,64,128])
SA2	SA1	(1024,0.05,[128,128,256])
SA3	SA2	(512,0.1,[128,128,256])
SA4	SA3	(256,0.2,[128,128,256])
FP1	SA3, SA4	[256,256]
FP2	SA2, SA3	[256,256]

dataset [3] for several reasons. Firstly, GraspNet is widely utilized in recent studies, enabling direct comparisons with state-of-the-art methods. Secondly, it offers an extensive collection of diverse objects, encompassing variations in shape, size, and material properties. Thirdly, the dataset provides rich annotations and labels, including grasp points and object attributes, essential for training and evaluating our algorithm effectively. Furthermore, GraspNet features real-world scenes, capturing environmental complexities and occlusions commonly encountered in practical robotic applications. Lastly, its availability to the public promotes collaboration within the research community, facilitating advancements in robotic grasping technology.

A. DATASET

We conduct evaluations and comparisons on the publicly available GraspNet-1Billion dataset [3]. This dataset consists of 97,280 RGBD images captured in 190 cluttered scenes, offering a vast repository of over one billion grasp poses for 88 distinct objects. The objects in these scenes vary in shape, texture, size, material, and occlusion conditions, presenting an ideal benchmark to gauge our model's generalization capacity and robustness in the face of occlusions. Each object in the dataset is annotated with an accurate 3D mesh model, camera poses, 6D object poses, object masks, and bounding boxes for all frames. This rich annotation enables the straightforward generation of ground truth votes and grasp configurations. Following [3], we partitioned the dataset into distinct training and testing sets. Specifically, 100 scenes were designated for training purposes, while the remaining 90 scenes were set aside for testing. To further assess the model's generalizability, the test dataset is stratified into subsets: scenes featuring novel objects, scenes with previously unseen yet similar objects, and scenes containing objects encountered during training. This deliberate partitioning allows for a comprehensive evaluation of our model's performance across a spectrum of diverse scenarios.

B. IMPLEMENTATION DETAILS

In our implementation, we utilize a pre-trained ResNet34 model, pretrained on the ImageNet dataset, as the encoder for RGB images. The output appearance feature from this architecture comprises 256 channels. For point cloud feature extraction, we randomly sample 12,288 points from depth images and employ a PointNet++ [49]-based feature learning network, which also yields a 256-channel output. The detailed layer parameters of PointNet++ [49] are presented in Table 2. In the voting and context learning modules, we form $K = 128$ clusters, producing a new feature map $\mathcal{F}_{context} \in 128 \times 512$. Subsequently, 128 grasps are generated from this new feature map. The prediction layer comprises $5 + V + 2A$ channels, with $V = 120$ and $A = 6$. We set $\lambda 1 = \lambda_2 = 1.0$ and $\alpha = \beta = \gamma = 1.0$. Our network is trained entirely using a batch size of 8 and optimized with Adam, employing a learning rate of 0.001 for 200 epochs. Training on a single Nvidia GeForce RTX 2080 Ti 11GB GPU takes approximately 20 hours. Figure 4 shows learning curves.

C. EVALUATION ON GRASPNET-1BILLION

Evaluation metric: We adopt *Precision@k* [3] as a key measure to evaluate the accuracy of our predicted grasp poses, particularly focusing on the top-ranked predictions. For each predicted grasp pose G_i, we determine its correctness by associating it with the target object based on the point cloud inside the gripper and considering the force-closure metric [72] under different friction coefficients μ. *Precision@k* measures the precision of the top-k ranked grasps. It quantifies the ratio of correct grasps among the top-k predicted grasps. To compute *Precision@k*, we rank the predicted grasp poses based on their confidence scores, select the top-k ranked grasp poses, and evaluate the precision of these top-k grasps by determining the proportion of true positive grasps among them. To provide a comprehensive evaluation, we compute the Average *Precision@k* (AP_k) across a range of values for k, specifically ranging from 1 to 50. The AP_k metric is crucial for assessing the accuracy of our grasp detection algorithm in cluttered scenes, where prioritizing the precision of top-ranked grasps is essential for successful robotic manipulation. We report the results of AP_k in our experimental evaluation, including comparisons across different scenarios and friction coefficients μ, allowing us to demonstrate the robustness and effectiveness of our 6-DoF grasp detection approach.

Figures 5, 6 and 7 show some qualitative results from the baseline VoteGrasp [2] and the proposed method. Tables 3 and 4 demonstrate the performance comparison between our approach and state-of-the-art methods [1], [1], [3], [7], [11], [12], [66], [68], [69], [71]. We have chosen to compare our method with these particular approaches because they are considered state-of-the-art in vision-based robot grasping. Their implementations are publicly available. Additionally, all of these methods are capable of generating grasp configurations suitable for a parallel-jaw gripper, which aligns with the focus of our research. By benchmarking our approach against these established methods, we aim to provide a comprehensive evaluation of its performance and demonstrate its effectiveness in comparison to existing state-of-the-art techniques. For a fair comparison, we re-implemented and re-trained each method using the provided code and settings from their original papers. The table

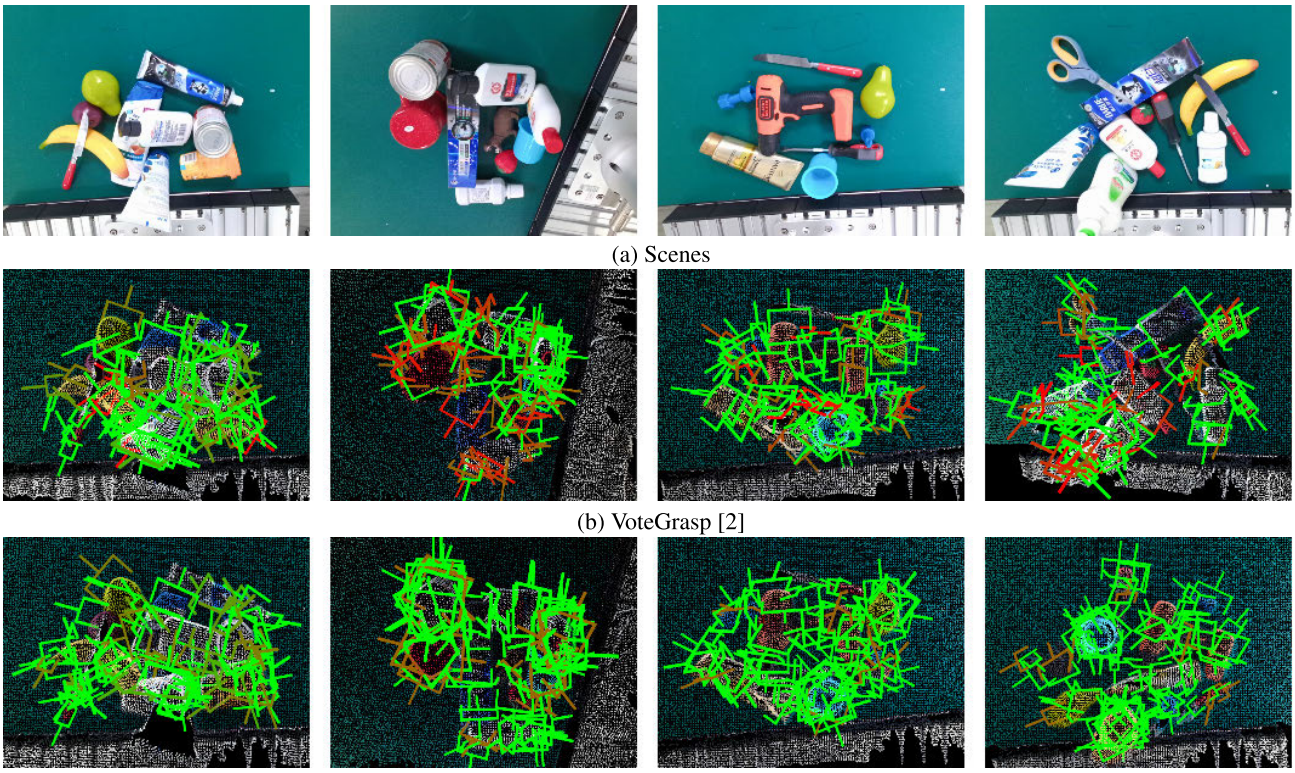


FIGURE 5. Examples of input scenes and predicted grasps from VoteGrasp [2] and the proposed method. Grasps are color-coded using an RGB scale, with red indicating lower confidence or quality and green indicating higher confidence or quality. Performance disparities between VoteGrasp and the proposed method are discernible, showcasing contrasting performance in equivalent scenarios.

TABLE 3. The table shows the results on GraspNet-1Billion test set captured by RealSense sensors.

	Seen				Unseen (but similar)				Novel			
	AP	$AP_{0.8}$	$AP_{0.6}$	$AP_{0.4}$	*AP*	$AP_{0.8}$	$AP_{0.6}$	$AP_{0.4}$	*AP*	$AP_{0.8}$	$AP_{0.6}$	$AP_{0.4}$
GG-CNN [66]	15.5	21.8	16.3/	10.3	13.3	18.4	13.9	4.6	5.5	5.9	5.7	1.9
Chu et al. [67]	16.0	23.7	16.5	10.8	15.4	20.2	15.9	7.1	7.6	8.7	7.8	2.5
GPD [7]	22.9	28.5	23.5	12.8	21.3	27.8	21.8	9.6	8.2	8.9	8.7	2.7
PointNetGPD [11]	26.0	33.0	26.6	15.4	22.7	29.2	23.2	10.8	9.2	9.9	9.6	2.7
Fang et al. [3]	27.6	33.4	27.8	17.0	26.1	34.2	26.6	14.2	10.6	11.3	10.9	4.0
Wang et al. [68]	29.2	35.2	29.7	19.0	28.4	36.3	28.8	16.6	11.9	13.1	12.4	6.2
Gou et al. [12]	28.0	33.5	28.5	17.8	27.2	36.3	27.6	15.6	12.3	12.	12.5	5.6
Contact-GraspNet [1]	29.9	35.2	31.4	19.5	28.2	37.0	28.6	16.3	13.2	13.5	13.6	6.8
Zheng et al. [69]	30.2	35.0	30.8	21.9	30.6	38.0	30.8	18.1	14.7	15.0	14.9	8.1
Chen et al. [70]	32.3	37.1	32.8	21.8	31.1	39.4	31.6	18.7	15.4	16.7	15.8	8.1
Zheng et al. [71]	33.0	37.2	33.5	23.4	32.1	39.2	32.4	19.6	15.4	16.1	15.6	9.2
VoteGrasp [2]	34.1	38.9	34.9	24.0/	33.0	40.8	33.3	20.5	16.9	17.5	17.2	10.0
Ours	**38.5**	**43.1**	**38.8**	**29.3**	**37.2**	**44.1**	**37.6**	**25.1**	**21.5**	**22.5**	**21.8**	**12.6**

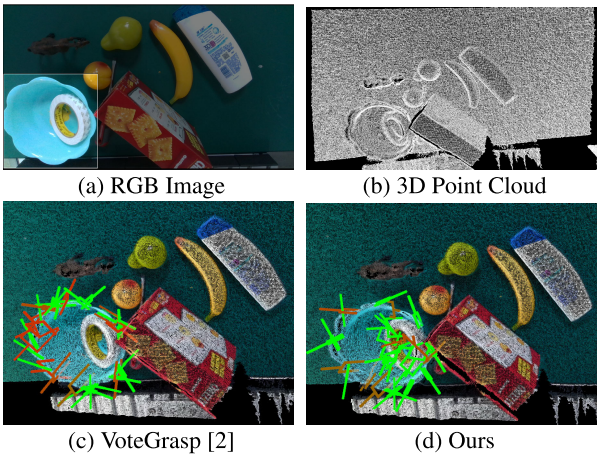
showcases the evaluation outcomes categorized into "Seen," "Unseen (but similar)," and "Novel" objects, aiding in assessing the model's generalization capability.

For the RealSense sensor data, our method consistently outperforms existing approaches across all evaluation metrics. Notably, our approach achieves an impressive Average Precision (*AP*) of 38.5 for seen objects, 37.2 for unseen (but similar) objects, and 21.5 for novel objects. These results surpass the performance of the baseline method VoteGrasp [2] in all categories, showcasing the efficacy of the proposed enhancements. When considering precision at various thresholds ($AP_{0.8}$, $AP_{0.6}$, and $AP_{0.4}$), our method consistently maintains superior performance, indicating its ability to generate more accurate and reliable grasps across different scenarios.

Similar trends are observed in the evaluation results for the Kinect sensor data. Our method outshines existing approaches, including the baseline method, across all metrics. The *AP* values for seen, unseen (but similar), and novel objects are 39.2, 38.0, and 21.2, respectively, demonstrating the robustness and generalization capabilities of our approach. The precision at various overlap thresholds further

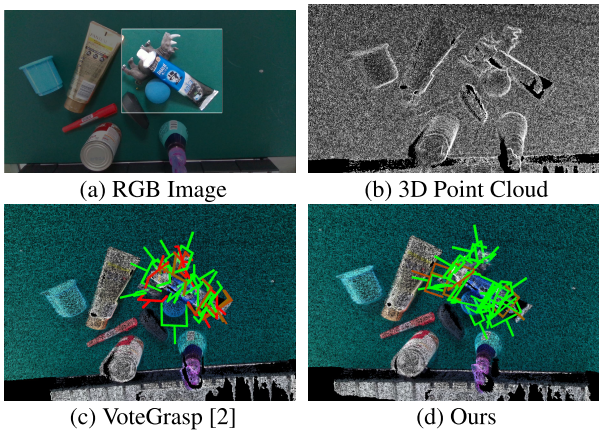
TABLE 4. The table shows the results on GraspNet-1Billion test set captured by Kinect sensors.

	Seen				Unseen (but similar)				Novel			
	AP	$AP_{0.8}$	$AP_{0.6}$	$AP_{0.4}$	AP	$AP_{0.8}$	$AP_{0.6}$	$AP_{0.4}$	AP	$AP_{0.8}$	$AP_{0.6}$	$AP_{0.4}$
GG-CNN [66]	16.9	22.5	17.1	11.2	15.1	19.8	15.6	6.2	7.4	8.8	7.8	1.3
Chu et al. [67]	17.6	24.7	17.9	12.7	17.4	21.6	17.5	8.9	8.0	9.3	8.2	1.8
GPD [7]	24.4	30.2	24.8	13.5	23.2	28.6	23.7	11.3	9.6	10.1	9.9	3.2
PointNetGPD [11]	27.6	34.2	28.3	17.8	24.4	30.8	24.8	12.8	10.7	11.2	10.9	3.2
Fang et al. [3]	29.9	36.2	30.6	19.3	27.8	33.2	28.1	16.6	11.5	12.9	11.7	3.6
Wang et al. [68]	31.2	38.0	31.8	20.9	29.9	35.5	31.6	18.8	13.4	14.6	13.8	5.8
Gou et al. [12]	32.1	39.5	32.5	20.9	30.4	37.9	30.5	18.7	13.1	13.8	13.4	6.0
Contact-GraspNet [1]	31.4	39.0	31.8	21.6	29.0	35.2	29.9	18.9	13.9	14.7	14.4	7.7
Zheng et al. [69]	34.2	41.5	34.9	25.1	32.3	41.2	32.6	21.8	15.8	15.8	15.8	8.3
Chen et al. [70]	33.5	42.4	33.9	23.9	32.0	37.7	32.6	20.9	16.2	17.9	16.6	9.2
Zheng et al. [71]	36.1	44.0	36.6	26.0	34.6	42.0	34.9	23.2	17.2	17.8	17.5	9.9
VoteGrasp [2]	37.5	45.6	37.6	27.7	35.9	43.3	36.4	24.7	18.5	18.9	18.7	10.6
Ours	**39.2**	**46.7**	**39.5**	**30.8**	**38.0**	**45.2**	**38.8**	**28.1**	**21.2**	**22.9**	**21.4**	**13.2**



(a) RGB Image (b) 3D Point Cloud (c) VoteGrasp [2] (d) Ours

FIGURE 6. An illustrative example highlighting the performance disparity between our proposed method and the baseline approach. In this scenario, the presence of tape inside the bowl complicates object distinction. VoteGrasp, lacking color information, fails to generate grasps for the tape, whereas our method, leveraging fusion of appearance and geometry information, comprehends the contextual cues and successfully generates grasps for both objects.



(a) RGB Image (b) 3D Point Cloud (c) VoteGrasp [2] (d) Ours

FIGURE 7. An illustrative example showcasing the performance contrast between our proposed method and the baseline approach. In this scenario, the highlighted objects are closely clustered together, making it challenging for baseline methods to generate high-quality grasps due to the proximity and occlusion. Our method, however, adeptly leverages fusion of appearance and geometry information to provide reliable grasps even in complex scenarios, demonstrating its superior performance in handling close proximity and occlusion challenges.

reinforces the effectiveness of our method in generating high-quality grasps. With $AP_{0.8}$, $AP_{0.6}$, and $AP_{0.4}$ values consistently surpassing those of other methods, our approach excels in providing grasps that exhibit strong alignment with ground truth annotations.

These results suggest that the proposed method not only achieves superior performance on familiar objects but also demonstrates a remarkable ability to generalize to unseen and novel objects. The inclusion of advanced feature learning components in our methodology contributes to its success in generating accurate and robust grasps across diverse scenarios.

Inference Time: Our experiments were conducted on an Intel Xeon E-2716G CPU clocked at 3.7 GHz, paired with an Nvidia GeForce RTX 2080 Ti GPU featuring 11GB of memory. The runtime analysis of all evaluated methods is graphically represented in Figure 8. Our approach achieves a runtime of 100 ms per RGBD image. This fine balance between accuracy and speed empowers our method to proficiently generate grasp configurations in cluttered scenes, rendering it well-suited for diverse real-world scenarios.

TABLE 5. Ablation study on the GraspNet-1Billion test set captured by RealSense and Kinect sensors. Performance comparison includes our method without Visual-Guided 3D Geometric Feature Learning (Ours (-VGG)), our method without Geometric-Guided Visual Feature Learning (Ours (-GGV)), our method with all components enabled (Ours (Full)), and our previous work with the same voting module [2] (VoteGrasp). Evaluation results are reported for seen, unseen (but similar), and novel object categories with *AP* metric.

	Seen	Unseen (but similar)	Novel
VoteGrasp [2]	36.0	34.5	17.3
Ours (-VGG)	34.2	32.7	14.6
Ours (-GGV)	33.7	31.3	14.5
Ours (Full)	**38.7**	**37.6**	**21.3**

D. ABLATION STUDY

We conduct an ablation study on the GraspNet-1Billion test set, captured using RealSense and Kinect sensors, to assess the impact of different components in our proposed

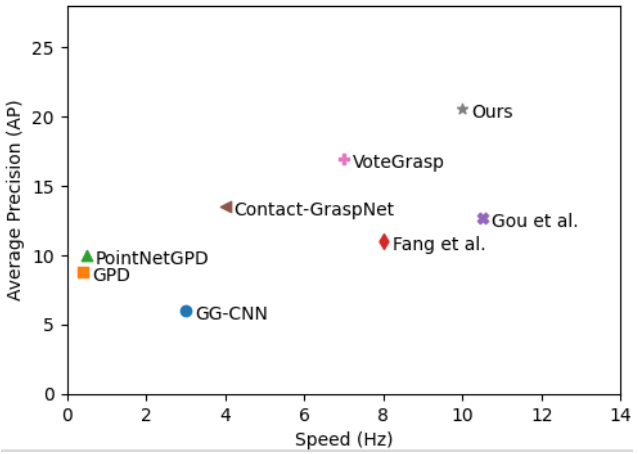


FIGURE 8. Comparison of running speed (Hz) and AP on GraspNet-1Billion dataset.

method. Table 5 presents the performance comparison across various configurations. Our previous work, VoteGrasp [2], achieved *AP* values of 36.0 for seen objects, 34.5 for unseen (but similar) objects, and 17.3 for novel objects. In the ablation study, Ours (-VGG) and Ours (-GGV) represent our method without Visual-Guided 3D Geometric Feature Learning and without Geometric-Guided Visual Feature Learning, respectively. Ours (-VGG) reports *AP* values of 34.2, 32.7, and 14.6, while Ours (-GGV) reports values of 33.7, 31.3, and 14.5 for seen, unseen (but similar), and novel objects, respectively. The full configuration of our method, Ours (Full), demonstrates superior performance across all categories, achieving *AP* values of 38.7, 37.6, and 21.3 for seen, unseen (but similar), and novel objects, respectively. This emphasizes the synergistic effect of integrating Visual-Guided 3D Geometric Feature Learning and Geometric-Guided Visual Feature Learning, leading to a more robust and effective grasp generation pipeline.

The distinctions between Ours (-VGG) and Ours (-GGV) when compared to our complete approach Ours (Full) showcase the significant impact of these modules. When excluding the VGG module, the method lacks the ability to effectively integrate RGB features, leading to a considerable reduction in performance across all evaluation categories: Seen, Unseen (but similar), and Novel objects. Similarly, without the GGV module, the model fails to appropriately fuse depth-based features, resulting in notable performance degradation in grasp detection across the board. The performance drop in both cases reaffirms the critical role played by these modules in amalgamating complementary information from RGB and depth data. It highlights their significance in capturing nuanced visual cues from different sources, which are vital for accurate and robust grasp detection. This clear decline in performance underlines the necessity of the VGG and GGV modules in our model's architecture, demonstrating their collective contribution to a more comprehensive understanding of the scene by integrating information from diverse

TABLE 6. Results of real robot experiments. The networks were trained on the GraspNet-1Billion dataset. The table shows the number of attempts, the number of successful attempts, and the grasp success rate.

Method	Attempt	Success	Success Rate
GPD [7]	200	134	67%
PointNetGPD [11]	200	130	65%
Fang et al. [3]	200	150	75%
Gou et al. [12]	200	152	76%
Contact-GraspNet [1]	200	154	77%
VoteGrasp [2]	200	156	78%
Ours	200	**168**	**84**%

modalities. The substantial discrepancy in results showcases that these modules are not just supplementary but rather pivotal components in leveraging the combined strengths of RGB and depth information. Their absence leads to a significant loss in the model's ability to discern crucial features necessary for precise grasp detection, emphasizing the vital role of these fusion modules in enhancing the model's performance across various object scenarios.

E. ROBOTIC GRASPING EXPERIMENT

The experiments were conducted with a Franka Emika Panda robot arm with 7-DOF, equipped with a parallel-jaw gripper as shown in Figure 9. To capture RGBD data, we used either ASUS Xtion PRO LIVE sensor or Microsoft Kinect sensor v2. The whole system is implemented using the ROS and MoveIt! frameworks.

To ensure a robust and comprehensive evaluation, we implemented a series of carefully considered factors to standardize experimental conditions and facilitate fair comparisons among the tested grasp detection methods. Firstly, we adopted a standardized object set comprising various shapes, sizes, materials, and weights representative of the objects commonly encountered in the robot's intended application domain. This consistent set of objects ensured that each method was evaluated on a diverse range of manipulation tasks, enhancing the validity of the comparisons. In addition, we randomized the initial poses and orientations of the objects in the scene for each trial. Objects were placed in different configurations, including upright, tilted, and partially occluded positions, to simulate the variability of real-world scenarios. By randomizing object poses and orientations, we introduced a degree of unpredictability that closely mimicked the challenges encountered in practical robotic manipulation tasks. Furthermore, we maintained a consistent number of objects in each trial to ensure comparability between different grasping methods. Whether testing with a single object, multiple objects, or cluttered scenes, the number of objects remained constant across experiments, enabling fair assessments of performance. Consistency in environmental conditions, including lighting, background textures, and workspace cleanliness, was crucial to maintaining experimental integrity. By ensuring these factors remained constant throughout the experiments,

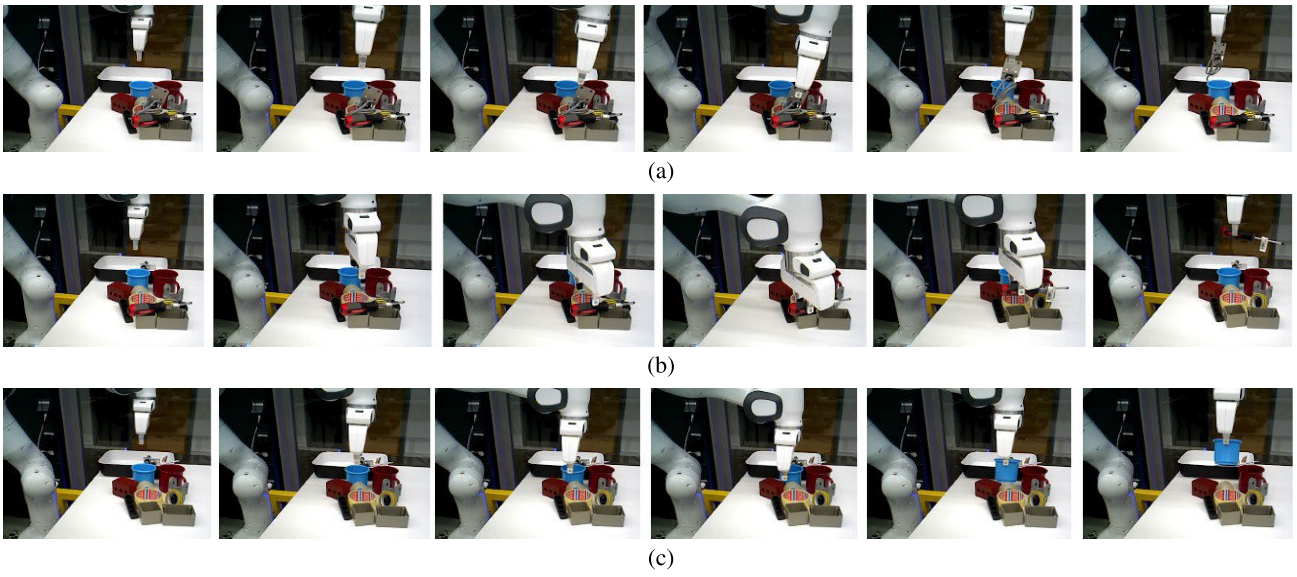


FIGURE 9. Examples of input scenes and predicted grasps from VoteGrasp [2] and the proposed method. The different intensity of grasp color denotes the confidence score of grasps. Green refers to the highest quality grasps and red refers to the lowest ones.

we minimized external influences that could impact the robot's perception and grasping performance. Moreover, we kept the robot configuration, including arm configuration and gripper type, consistent throughout the experiments. This approach isolated the effect of grasping methods and enabled direct comparisons between different approaches. To account for variability and ensure statistical significance in the results, we employed randomization and replication techniques. The order of experiments was randomized, and each experiment was replicated multiple times.

Specifically, we conducted a real-world evaluation of state-of-the-art grasp detection methods, each trained on the GraspNet-1Billion dataset for a fair comparison. Furthermore, we implemented measures to standardize the experimental setup, including the careful selection of objects to match the gripper's shapes and sizes. This approach ensured that each method encountered objects that were representative of real-world manipulation tasks, contributing to the validity of the comparisons. Moreover, in each experimental scenario, we deliberately arranged a random subset of 10-15 objects on the table in a haphazard manner. This arrangement aimed to mimic the unpredictability and variability inherent in real-world environments, thus providing a challenging yet realistic testing scenario for the evaluated methods. By randomizing the objects' positions and orientations, we further enhanced the fairness and robustness of the comparisons, ensuring that each method faced similar challenges and scenarios during evaluation. Throughout the experiments, each method underwent 200 grasp attempts, with the robot randomly selecting objects for interaction. A grasp was considered successful only if the robot could grasp and lift the object within a single attempt, reflecting the practical requirement for efficient and reliable manipulation. This strict evaluation criterion maintained consistency across the experiments and enabled a clear assessment of each method's performance under real-world conditions. The results in Table 6 demonstrate our method's superiority, achieving an 84% success rate outperforming all other methods. This highlights the proposed framework's efficacy in real-world grasping scenarios, attributing the increased success rate to the integration of estimated depth data, underscoring the significance of richer input data for precise and effective.

We further demonstrate the adaptability and ease of modification that our new loss function (Eq. 15) brings to our method in real-world applications, we conducted experiments in which specific objects were designated for robot manipulation. We performed 100 attempts at grasping each of specified objects and recorded the success rates as shown in Table 7. It is clear from the table that our method outperforms state-of-the art model-based methods also known as 6D object pose estimation. The results suggest that the proposed approach for grasp generation is more robust and generalizable than 6D pose-based grasp generation methods, making it suitable for target object grasping applications.

The success demonstrated in our experimental evaluations underscores the potential impact of our method in advancing the field of robotic manipulation. However, the current study represents a stepping stone, and future work could explore several promising directions. One avenue for further investigation is the extension of our approach to handle dynamic and cluttered environments. Adapting the model to dynamically changing scenes and improving its robustness in cluttered scenarios would be crucial for real-world deployment. Moreover, exploring the integration of additional sensory information, such as tactile feedback or

TABLE 7. Comparison of our method with 6D object pose estimation approaches in real robot grasping. The networks were trained on the GraspNet-1Billion dataset. The table shows the number of attempts, the number of successful attempts, and the grasp success rate.

Method	Attempt	Success	Success Rate
Wang et al. [13]	100	78	78%
He et al. [73]	100	79	79%
Kumar et al. [74]	100	76	76%
Hu et al. [75]	100	73	73%
Liu et al. [76]	100	78	78%
VoteGrasp [2]	100	78	78%
Ours	100	**83**	**83**%

proprioceptive data, could further enhance the versatility and adaptability of the grasp generation system. This multi-modal integration might enable the model to refine grasp configurations in response to real-time environmental changes and uncertainties. Additionally, addressing the transferability of the trained model to various robotic platforms and scenarios remains a vital aspect of future research. Investigating methods to facilitate domain adaptation and transfer learning could enable the seamless deployment of our approach in diverse robotic applications, ranging from industrial settings to domestic environments.

V. CONCLUSION

In this study, we addressed the fundamental challenge of grasp generation in robotic manipulation by introducing an innovative approach that bypasses the need for specialized depth sensors. Our method revolutionizes grasp generation by leveraging tailored deep learning techniques to estimate depth from color (RGB) images directly. This paradigm shift allows the computation of predicted point clouds solely from RGB inputs, eliminating the dependency on traditional depth sensors. A pivotal contribution lies in the development of a fusion module adept at seamlessly integrating features derived from RGB images with those inferred from predicted point clouds. This fusion process harnesses the strengths of both modalities, significantly enhancing grasp configurations. Our experimental evaluations unequivocally validate the efficacy of our approach, demonstrating its superiority in generating grasp configurations compared to existing methods. We achieved a remarkable 4% improvement in average precision compared to state-of-the-art grasp detection methods. Moreover, in real robot grasping experiments, our proposed method exhibited a 6% increase in success rate compared to state-of-the-art grasp detection methods and a 5% improvement compared to object pose estimation frameworks. Our future work will focus on extending the capabilities of the proposed approach to handle dynamic, cluttered environments, exploring multi-modal integration for improved adaptability, and ensuring the transferability of the model across different robotic platforms and scenarios. These endeavors hold the promise of further elevating the versatility, adaptability, and real-world applicability of grasp generation in robotics.

REFERENCES

[1] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-GraspNet: Efficient 6-DoF grasp generation in cluttered scenes," 2021, *arXiv:2103.14127*.
[2] D.-C. Hoang, J. A. Stork, and T. Stoyanov, "Context-aware grasp generation in cluttered scenes," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 1492–1498.
[3] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "GraspNet-1Billion: A large-scale benchmark for general object grasping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11441–11450.
[4] D.-C. Hoang, L.-C. Chen, and T.-H. Nguyen, "Sub-OBB based object recognition and localization algorithm using range images," *Meas. Sci. Technol.*, vol. 28, no. 2, Feb. 2017, Art. no. 025401.
[5] G. Du, K. Wang, S. Lian, and K. Zhao, "Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: A review," *Artif. Intell. Rev.*, vol. 54, no. 3, pp. 1677–1734, Mar. 2021.
[6] D.-C. Hoang, J. A. Stork, and T. Stoyanov, "Voting and attention-based pose relation learning for object pose estimation from 3D point clouds," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 8980–8987, Oct. 2022.
[7] A. T. Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *Int. J. Robot. Res.*, vol. 36, nos. 13–14, pp. 1455–1473, Dec. 2017.
[8] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from RGBD images: Learning using a new rectangle representation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3304–3311.
[9] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Robot. Res.*, vol. 34, nos. 4–5, pp. 705–724, Apr. 2015.
[10] D.-C. Hoang, A.-N. Nguyen, V.-D. Vu, D.-Q. Vu, V.-T. Nguyen, T.-U. Nguyen, C.-T. Tran, K.-T. Phan, and N.-T. Ho, "Grasp configuration synthesis from 3D point clouds with attention mechanism," *J. Intell. Robotic Syst.*, vol. 109, no. 3, p. 71, Nov. 2023.
[11] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "PointNetGPD: Detecting grasp configurations from point sets," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 3629–3635.
[12] M. Gou, H.-S. Fang, Z. Zhu, S. Xu, C. Wang, and C. Lu, "RGB matters: Learning 7-DoF grasp poses on monocular RGBD images," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 13459–13466.
[13] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3338–3347.
[14] D.-C. Hoang, A.-N. Nguyen, V.-D. Vu, T.-U. Nguyen, D.-Q. Vu, P.-Q. Ngo, N.-A. Hoang, K.-T. Phan, D.-T. Tran, V.-T. Nguyen, Q.-T. Duong, N.-T. Ho, C.-T. Tran, V.-H. Duong, and A.-T. Mai, "Graspability-aware object pose estimation in cluttered scenes," *IEEE Robot. Autom. Lett.*, vol. 9, no. 4, pp. 3124–3130, Apr. 2024.
[15] V.-D. Vu, D.-D. Hoang, P. X. Tan, V.-T. Nguyen, T.-U. Nguyen, N.-A. Hoang, K.-T. Phan, D.-T. Tran, D.-Q. Vu, P.-Q. Ngo, Q.-T. Duong, A.-N. Nguyen, and D.-C. Hoang, "Occlusion-robust pallet pose estimation for warehouse automation," *IEEE Access*, vol. 12, pp. 1927–1942, 2024.
[16] D.-C. Hoang, A. J. Lilienthal, and T. Stoyanov, "Panoptic 3D mapping and object pose estimation using adaptively weighted semantic information," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1962–1969, Apr. 2020.
[17] U. Asif, M. Bennamoun, and F. A. Sohel, "RGB-D object recognition and grasp detection using hierarchical cascaded forests," *IEEE Trans. Robot.*, vol. 33, no. 3, pp. 547–564, Jun. 2017.
[18] Y. Song, J. Wen, D. Liu, and C. Yu, "Deep robotic grasping prediction with hierarchical RGB-D fusion," *Int. J. Control, Autom. Syst.*, vol. 20, no. 1, pp. 243–254, Jan. 2022.
[19] D.-C. Hoang, A. J. Lilienthal, and T. Stoyanov, "Object-RPE: Dense 3D reconstruction and pose estimation with convolutional neural networks," *Robot. Auto. Syst.*, vol. 133, Nov. 2020, Art. no. 103632.
[20] H. Tian, K. Song, S. Li, S. Ma, and Y. Yan, "Lightweight pixel-wise generative robot grasping detection based on RGB-D dense fusion," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.
[21] Y. Wu, Y. Fu, and S. Wang, "Real-time pixel-wise grasp detection based on RGB-D feature dense fusion," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2021, pp. 970–975.
[22] D.-C. Hoang, T. Stoyanov, and A. J. Lilienthal, "Object-RPE: Dense 3D reconstruction and pose estimation with convolutional neural networks for warehouse robots," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Sep. 2019, pp. 1–6.

[23] V. Arampatzakis, G. Pavlidis, N. Mitianoudis, and N. Papamarkos, "Monocular depth estimation: A thorough review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 4, pp. 2396–2414, Apr. 2024.

[24] M. Ng, Y. Huo, H. Yi, Z. Wang, J. Shi, Z. Lu, and P. Luo, "Learning depth-guided convolutions for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 4306–4315.

[25] X. Lou, Y. Yang, and C. Choi, "Learning to generate 6-DoF grasp poses with reachability awareness," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 1532–1538.

[26] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1316–1322.

[27] M. Kokic, D. Kragic, and J. Bohg, "Learning task-oriented grasping from human activity datasets," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3352–3359, Apr. 2020.

[28] P. Schmidt, N. Vahrenkamp, M. Wächter, and T. Asfour, "Grasping of unknown objects using deep convolutional neural networks based on depth images," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 6831–6838.

[29] D. Yang, T. Tosun, B. Eisner, V. Isler, and D. Lee, "Robotic grasping through combined image-based grasp proposal and 3D reconstruction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 6350–6356.

[30] P. Ni, W. Zhang, X. Zhu, and Q. Cao, "PointNet++ grasping: Learning an end-to-end spatial grasp generation algorithm from sparse point clouds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3619–3625.

[31] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proc. Millennium Conf. IEEE Int. Conf. Robot. Automat. (ICRA) Symposia*, vol. 1. IEEE, Apr. 2000, pp. 348–353.

[32] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 2. IEEE, Sep. 2003, pp. 1824–1829.

[33] A. Saxena, S. Chung, and A. Ng, "Learning depth from single monocular images," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 18, 2005. [Online]. Available: Link: https://papers.nips.cc/paper_files/paper/2005/hash/17d8da815fa21c57af9829fb0a869602-Abstract.html

[34] A. Saxena, M. Sun, and A. Y. Ng, "Make3D: Learning 3D scene structure from a single still image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 824–840, May 2009.

[35] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," 2014, *arXiv:1406.2283*.

[36] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2650–2658.

[37] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 239–248.

[38] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, "Multi-scale continuous CRFs as sequential deep networks for monocular depth estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 161–169.

[39] J. Jiao, Y. Cao, Y. Song, and R. Lau, "Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 53–69.

[40] M. Ramamonjisoa and V. Lepetit, "SharpNet: Fast and accurate recovery of occluding contours in monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 2109–2118.

[41] Z. Zhang, Z. Cui, C. Xu, Y. Yan, N. Sebe, and J. Yang, "Pattern-affinitive propagation across depth, surface normal and semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4101–4110.

[42] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," 2019, *arXiv:1907.10326*.

[43] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7482–7491.

[44] W. Yin, Y. Liu, C. Shen, and Y. Yan, "Enforcing geometric constraints of virtual normal for depth prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5683–5692.

[45] T. Chen, S. An, Y. Zhang, C. Ma, H. Wang, X. Guo, and W. Zheng, "Improving monocular depth estimation by leveraging structural awareness and complementary datasets," in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 2020, pp. 90–108.

[46] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12159–12168.

[47] G. Yang, H. Tang, M. Ding, N. Sebe, and E. Ricci, "Transformer-based attention networks for continuous pixel-wise prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16249–16259.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[49] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," 2017, *arXiv:1706.02413*.

[50] X. Chen, K. Y. Lin, J. Wang, W. Wu, C. Qian, H. Li, and G. Zeng, "Bi-directional cross-modality feature propagation with separation-and-aggregation gate for RGB-D semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Aug. 2020, pp. 561–577.

[51] D. Seichter, M. Köhler, B. Lewandowski, T. Wengefeld, and H.-M. Gross, "Efficient RGB-D semantic segmentation for indoor scene analysis," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 13525–13531.

[52] Y. Wang, X. Chen, L. Cao, W. Huang, F. Sun, and Y. Wang, "Multimodal token fusion for vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12176–12185.

[53] X. Liu, P. Ren, Y. Chen, C. Liu, J. Wang, H. Sun, Q. Qi, and J. Wang, "Sample-adapt fusion network for RGB-D hand detection in the wild," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[54] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, "TransFusion: Robust LiDAR-camera fusion for 3D object detection with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1080–1089.

[55] Z. Chen, Z. Li, S. Zhang, L. Fang, Q. Jiang, and F. Zhao, "Deformable feature aggregation for dynamic multi-modal 3D object detection," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2022, pp. 628–644.

[56] Y. Kim, K. Park, M. Kim, D. Kum, and J. Won Choi, "3D dual-fusion: Dual-domain dual-query camera-LiDAR fusion for 3D object detection," 2022, *arXiv:2211.13529*.

[57] X. Chen, T. Zhang, Y. Wang, Y. Wang, and H. Zhao, "FUTR3D: A unified sensor fusion framework for 3D detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2023, pp. 172–181.

[58] X. Li, T. Ma, Y. Hou, B. Shi, Y. Yang, Y. Liu, X. Wu, Q. Chen, Y. Li, Y. Qiao, and L. He, "LoGoNet: Towards accurate 3D object detection with local-to-global cross-modal fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 17524–17534.

[59] L. Piccinelli, C. Sakaridis, and F. Yu, "IDisc: Internal discretization for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 21477–21487.

[60] Z. Wu, Y. Gan, L. Wang, G. Chen, and J. Pu, "MonoPGC: Monocular 3D object detection with pixel geometry contexts," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 4842–4849.

[61] S. Y. Kim, J. Zhang, S. Niklaus, Y. Fan, S. Chen, Z. Lin, and M. Kim, "Layered depth refinement with mask guidance," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3845–3855.

[62] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16239–16248.

[63] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017. [Online]. Available: https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[64] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.

[65] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

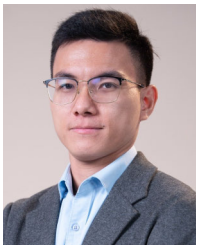
[66] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," 2018, *arXiv:1804.05172*.

[67] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3355–3362, Oct. 2018.

[68] C. Wang, H.-S. Fang, M. Gou, H. Fang, J. Gao, and C. Lu, "Graspness discovery in clutters for fast and accurate grasp detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15944–15953.

[69] L. Zheng, Y. Cai, T. Lu, and S. Wang, "VGPN: 6-DoF grasp pose detection network based on Hough voting," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 7460–7467.
[70] S. Chen, W. Tang, P. Xie, W. Yang, and G. Wang, "Efficient heatmap-guided 6-DoF grasp detection in cluttered scenes," *IEEE Robot. Autom. Lett.*, vol. 8, no. 8, pp. 4895–4902, Aug. 2023.
[71] L. Zheng, W. Ma, Y. Cai, T. Lu, and S. Wang, "GPDAN: Grasp pose domain adaptation network for sim-to-real 6-DoF object grasping," *IEEE Robot. Autom. Lett.*, vol. 8, no. 8, pp. 4585–4592, Aug. 2023.
[72] V.-D. Nguyen, "Constructing force-closure grasps," *Int. J. Robot. Res.*, vol. 7, no. 3, pp. 3–16, 1988.
[73] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "PVN3D: A deep point-wise 3D keypoints voting network for 6DoF pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11629–11638.
[74] A. Kumar, P. Shukla, V. Kushwaha, and G. C. Nandi, "Context-aware 6D pose estimation of known objects using RGB-D data," 2022, *arXiv:2212.05560*.
[75] Y. Hu, P. Fua, W. Wang, and M. Salzmann, "Single-stage 6D object pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2927–2936.
[76] C. Liu, W. Sun, K. Zhang, J. Liu, X. Zhang, and S. Fan, "Prior geometry guided direct regression network for monocular 6D object pose estimation," in *Proc. 41st Chin. Control Conf. (CCC)*, Jul. 2022, pp. 6241–6246.

PHAN XUAN TAN (Member, IEEE) received the B.E. degree in electrical-electronic engineering from the Military Technical Academy, Vietnam, the M.E. degree in computer and communication engineering from Hanoi University of Science and Technology, Vietnam, and the Ph.D. degree in functional control systems from Shibaura Institute of Technology, Japan. He is currently an Associate Professor with Shibaura Institute of Technology. His current research interests include deep learning for visual computing, image and video processing, computational light field, 3D view synthesis, multimedia quality of experience, and multimedia networking.



DINH-CUONG HOANG received the Ph.D. degree in computer science from Örebro University, Sweden, in 2021. He is currently a Lecturer with Greenwich Vietnam, FPT University. His research interests include the intersection of computer vision, robotics, and machine learning. He is particularly interested in topics involving autonomy for robots, with a focus on perception algorithms.

ANH-NHAT NGUYEN received the B.S. degree in computer science from Duy Tan University, Da Nang, Vietnam, in 2012, and the M.S. degree from Huazhong University of Science and Technology (HUST), China, in 2018. He is currently a Lecturer with FPT University, Vietnam. His research interests include image processing, information security, physical layer secrecy, radio-frequency energy harvesting, and wireless sensor networks.

VAN-THIEP NGUYEN is currently pursuing the B.S. degree in artificial intelligence with FPT University, Hanoi, Vietnam. His current research interests include manipulation of objects by robots and object recognition.

VAN-DUC VU is currently pursuing the B.S. degree in artificial intelligence with FPT University, Hanoi, Vietnam. His research interest includes computer vision.

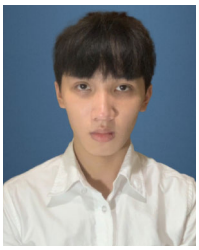
THU-UYEN NGUYEN is currently pursuing the B.S. degree in artificial intelligence with FPT University, Hanoi, Vietnam. Her research interest includes computer vision.

NGOC-ANH HOANG is currently pursuing the B.S. degree in artificial intelligence with FPT University, Hanoi, Vietnam. His research interest includes computer vision.

KHANH-TOAN PHAN is currently pursuing the B.S. degree in artificial intelligence with FPT University, Hanoi, Vietnam. His research interest includes computer vision.

DUC-THANH TRAN is currently pursuing the B.S. degree in artificial intelligence with FPT University, Hanoi, Vietnam. His research interest includes computer vision.

DUY-QUANG VU is currently pursuing the B.S. degree in artificial intelligence with FPT University, Hanoi, Vietnam. His research interest includes computer vision.



PHUC-QUAN NGO is currently pursuing the B.S. degree in computing with Greenwich Vietnam, FPT University, Hanoi, Vietnam. His research interests include computer vision and the Internet of Things (IoT).

QUANG-TRI DUONG, photograph and biography not available at the time of publication.

NGOC-TRUNG HO is currently pursuing the B.S. degree in artificial intelligence with FPT University, Hanoi, Vietnam. His research interests include computer vision and robotics

CONG-TRINH TRAN is currently pursuing the B.S. degree in artificial intelligence with FPT University, Hanoi, Vietnam. His research interest includes computer vision.

VAN-HIEP DUONG, photograph and biography not available at the time of publication.

ANH-TRUONG MAI, photograph and biography not available at the time of publication.