applied sciences

MDPI

Article

Grasp Detection Combining Self-Attention with CNN in Complex Scenes

Jinxing Niu [1], Shuo Liu [1], Hanbing Li [2], Tao Zhang [1] and Lijun Wang [1,*]

[1] School of Mechanical Engineering, North China University of Water Resources and Electric Power, Zhengzhou 450011, China; niujinxing@ncwu.edu.cn (J.N.); w7926657@gmail.com (S.L.); ztncwu@126.com (T.Z.)

[2] SongShan Laboratory, Zhengzhou 450000, China; anatoly-li@foxmail.com

* Correspondence: wljmb@163.com

Abstract: In this paper, we present a novel approach that subtly combines the transformer with grasping CNN to achieve more optimal grasps in complex real-life situations. The approach comprises two unique designs that effectively improve grasp precision in complex scenes. The first essential design uses self-attention mechanisms to capture contextual information from RGB images, boosting contrast between key object features and their surroundings. We precisely adjust internal parameters to balance accuracy and computing costs. The second crucial design involves building a feature fusion bridge that processes all one-dimensional sequence features at once to create an intuitive visual perception for the detection stage, ensuring a seamless combination of the transformer block and CNN. These designs eliminate noise features in complex backgrounds and emphasize graspable object features, providing valuable semantic data to the subsequent grasping CNN to achieve appropriate grasping. We evaluated the approach on the Cornell and VMRD datasets. According to the experimental results, our method achieves better performance than the original grasping CNN in single-object and multi-object scenarios, exhibiting 97.7% and 72.2% accuracy on the Cornell and VMRD grasp datasets using RGB, respectively.

Keywords: grasp detection; transformer; deep learning

check for updates

Citation: Niu, J.; Liu, S.; Li, H.; Zhang, T.; Wang, L. Grasp Detection Combining Self-Attention with CNN in Complex Scenes. *Appl. Sci.* **2023**, *13*, 9655. https://doi.org/10.3390/app13179655

Academic Editors: Dimitris Mourtzis and Zhengjun Liu

Received: 18 June 2023
Revised: 22 August 2023
Accepted: 24 August 2023
Published: 25 August 2023

1. Introduction

The implementation of grasping technology is crucial for the intelligent automation of robots. Achieving robust grasping requires performing scene sensations, motion planning, and execution control simultaneously for the robotic arm. Grasping tasks are frequently used in structured scenes. Despite this, accurately perceiving the target object in unstructured environments, such as complex backgrounds, and predicting rapid and precise grasping remain challenging problems.

In the past, traditional methods that studied the physical geometric models and kinematics of objects were used to determine grasping poses. These methods are not robust enough to be applied to real-world scenarios [1–3]. Recently, deep learning methods have shown promising outcomes in detecting grasps for robots. Deep convolutional neural networks can learn to extract features suitable for specific tasks by simplifying the grasp detection problem definition [4,5], thus circumventing the need for manual feature design. Recent researches focused on utilizing convolutional neural networks (CNNs) for grasp detection [6]. Although these methods show satisfactory performance on simple single-object grasp detection tasks (e.g., Cornell dataset), there is still significant potential for grasp detection performance in complex life scenes. As illustrated in Figure 1 involving object overlapping and cluttered patterns, the CNN-based grasping network [7] was unable to achieve appropriate poses when grasping a screwdriver or a stapler. There may be two main reasons why these features have not been fully explored yet. The first is that datasets containing them have not been proposed so far, and the other is that the inherent nature of CNNs limits grasp detection performance (e.g., smaller sensory fields, generalization capability) in real-life situations, which usually contain complex backgrounds with cluttered objects. Thus, in this work, we were particularly motivated to investigate grasping detection considering real-life scenarios and better generalization capabilities, using only existing public datasets.

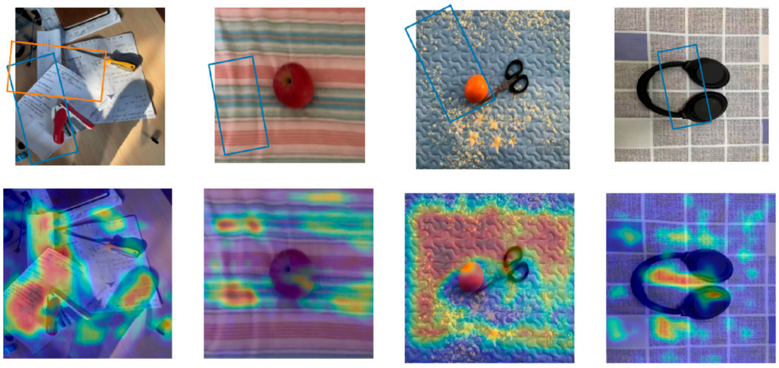

Figure 1. Examples of detection by grasping CNN [7].

In this work, a proposed grasping model combines transformer and CNN effectively, modeling both local and global perception, while emphasizing the distinction between graspable objects and complex backgrounds. The self-attention mechanism links information within image patches. Within our framework, the feature fusion bridge (FFB) captures discrete low-level features that are then aggregated into multi-scale high-level semantic features. High-level features are incorporated by the grasping CNN to determine the final grasping pose within complex backgrounds. Experimental results show that the algorithm has good performance in balancing accuracy and computing cost on popular grasping benchmark datasets, e.g., Cornell and VMRD. For complex backgrounds with strong interferences, our method shows much more superior grasp detection performance than the CNN-based method [7]. In summary, the main contributions are as follows:

We propose a combination of transformer and grasping CNN to be applied to predict grasps in complex backgrounds.

An effective feature fusion bridge is used to smooth the transition from the transformer to CNN, enabling multi-scale feature aggregation.

We evaluated our model on public benchmark datasets, Cornell and VMRD, and achieved excellent accuracy of 97.7% and 72.2%, respectively.

We collected images from real scenes to prove the effectiveness of the proposed method. Experimental results demonstrate that our model is able to make more-appropriate grasping judgments than the raw grasping CNN in complex scenes.

2. Related Work

To enable robots to determine optimal grasp angles and opening distances, it is necessary to have accurate modeling of the position, posture, and contour information of objects for precise grasp detection. Due to the constantly changing and complex characteristics of robot work environments, extracting and mapping relevant features is critical for effective object–background discrimination.

Earlier methods for grasp detection primarily relied on non-data-driven traditional algorithms, including analytical approaches. Such approaches analyze the surface properties of objects related to friction at contact points and apply geometry, kinematics, and dynamics to calculate corresponding grasps [2]. Despite their potential advantages, such approaches can be challenging to apply in real-world settings primarily due to their requirement for manually engineered features.

Recently, learning-based approaches have gained widespread attention and become the primary focus of research in grasp detection [4,5,8,9]. Supervised learning is employed to fit detection models in the dataset, allowing direct prediction of the grasp from the image [8,9], without the need to construct a three-dimensional model of the object. A new five-dimensional grasp rectangle representation is proposed in [4] as an alternative to the grasp point prediction model. This representation includes the grasp center point, the opening distance of the end effector, and the rotation angle, directly replacing the grasp representation based on three-dimensional space. The optimum grasp region is determined using a cascaded two-stage support vector machines (SVM) classifier. The deep learning method based on neural networks has achieved great advantages in image classification [10], and Lenz [5] introduced deep neural networks (DNNs) as the grasp detection classifier, designing a two-stage DNN to avoid manual feature design and improving the generalization ability of the model. Each potential grasp rectangle is evaluated and ranked. Based on the oriented rectangle, grasp detection is similar to object detection in computer vision, so many classic CNN structures [10–12] are applied in grasp detection research to improve the algorithm performance. In [6], Redmon proposed a one-stage grasp detection network based on AlexNet, treating the calculation of the grasp rectangle as a regression problem, and achieved feature extraction and grasp rectangle prediction evaluation solely based on object image information. Furthermore, S. Kumra [13] utilized ResNet-50 [11] to extract features on the image and used linear SVM as the prediction classifier to predict the object's grasp configuration from the features extracted from the last hidden layer of ResNet-50. Similarly, using CNN structures, ref. [7] encodes the input image's features with downsampling convolution layers, increases network depth and abstract generalization ability with ResNet modules, and decodes pixel-level grasp prediction with upsampling layers.

To summarize, existing research on grasp detection using deep learning technology primarily relies on commonly used CNN models for object detection like AlexNet, VGGNet, and ResNet. However, grasp detection and object detection differ significantly in their application scenarios. The former has more diverse application scenarios, and therefore, the grasp configuration of objects is complex and variable, requiring stricter parameters for grasp angle and position.

The VIT [14] replaces the traditional CNN model with the transformer [15] to extract image features. It proposes an end-to-end detection architecture that exhibits excellent performance in image classification tasks. The transformer has become a new paradigm in computer vision due to its exceptional ability to model long sequences and extract global features. Wang [16] demonstrates the feasibility of using the transformer for grasp detection tasks. It does so by proposing a transformer-based grasp detection model that utilizes an encoder–decoder architecture with skip connections.

Previous works have solely focused on grasping in normal scenes and not thoroughly investigated grasping in particular scenes. In such instances, it is essential to enhance the model's feature extraction and mapping abilities to distinguish objects from diverse environments accurately. This paper focuses on detecting grasps in complex backgrounds. We propose a hybrid method that utilizes both the grasping CNN and transformer for better performance, which has not yet been considered. We introduce self-attention to model global features based on the existing grasping CNN. This approach combines the benefits of each and achieves grasp detection in complex scenes with strong interferences. The results indicate that our method makes more accurate grasping judgments than the CNN model.

3. Problem Definition

For vision-based grasp detection, a visual sensor captures a multi-channel image that includes the object under consideration, while assuming the existence of multiple workable grasping configurations within the image. In this work, we employ an improved variant of the grasp rectangle representation as proposed in [17]. Specifically, in the case of a parallel-jaw gripper, a grasp rectangle is defined by the position and orientation of the end effector concerning the target object, as well as the quality of the grasp when executed.

$$\mathrm{G_R} = \{\mathrm{p}, \sigma_\mathrm{R}, \mathrm{w_R}, \mathrm{h_R}, \mathrm{q}\} \tag{1}$$

where p = {x, y, z} is the center coordinates, σ represents the clockwise rotation angle around the Z-axis, $(\mathrm{w_R}, \mathrm{h_R})$ is the grasp rectangle width and height in the robot frame where $\mathrm{w_R}$ also means the opening width of the end effector, and q is the grasp quality score.

The grasp representations in an image frame are given by:

$$\mathrm{G_i} = \{\mathrm{x_i}, \mathrm{y_i}, \sigma_\mathrm{i}, \mathrm{w_i}, \mathrm{h_i}, \mathrm{q}\} \tag{2}$$

where $\mathrm{x_i}$ and $\mathrm{y_i}$ denote the center coordinates of the grasp rectangle in the image frame, σ_i represents the rotation angle in the image frame, $(\mathrm{w_i}, \mathrm{h_i})$ denotes the grasp rectangle width and height in the image frame, and q is the same as in Equation (1).

Different from the 5-dimensional grasp in [4,6,13], the grasp quality q expresses the probability of a successful grasp, which is similar to the sample confidence in object detection. In detail, for each pixel, a floating number from 0 to 1, corresponding to the pixel position, is found and quantifies a grasping success, with values close to 1 indicating a higher likelihood. σ_i represents the rotation angle during grasping, with a range defined as $[-\frac{\pi}{2}, +\frac{\pi}{2}]$.

To execute a grasp, the map between the image frame and the robot frame needs to be established, as in the following:

$$\mathrm{G_R} = \mathrm{T_C^R}\left(\mathrm{T_i^C}(\mathrm{G_i})\right)$$

where

$$\mathrm{T_i^C} = \begin{bmatrix} \frac{1}{f} & 0 & 0 \\ 0 & \frac{1}{f} & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathrm{T_C^R} = \begin{bmatrix} \mathrm{R} & \mathrm{T} \end{bmatrix} \tag{3}$$

$\mathrm{T_i^C}$ is the transform matrix from image frame to camera frame, which is related to the focal length f of the camera. $\mathrm{T_C^R}$ is the transformation matrix from image frame to robot frame, consisting of a rotation matrix $\mathrm{R} \in \mathbb{R}^{3\times3}$ with a translation matrix $\mathrm{T} \in \mathbb{R}^{3\times1}$. $\mathrm{T_i^C}$ is obtained by camera calibration [18] and $\mathrm{T_C^R}$ by hand–eye calibration [19].

In the implementation, we calculate each pixel point of the grasping rectangle to obtain the position in the robot frame. The specific calculation process is as follows:

$$\begin{gathered} \begin{bmatrix} \mathrm{X_C} \\ \mathrm{Y_C} \\ \mathrm{Z_C} \end{bmatrix} = \mathrm{T_i^C} * \begin{bmatrix} \mathrm{x_i} \\ \mathrm{y_i} \\ 1 \end{bmatrix} * \mathrm{Z_C} = \begin{bmatrix} \frac{1}{f} & 0 & 0 \\ 0 & \frac{1}{f} & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \mathrm{x_i} \\ \mathrm{y_i} \\ 1 \end{bmatrix} * \mathrm{Z_C} \\ \begin{bmatrix} \mathrm{X_R} \\ \mathrm{Y_R} \\ \mathrm{Z_R} \end{bmatrix} = \mathrm{R} * \begin{bmatrix} \mathrm{X_C} \\ \mathrm{Y_C} \\ \mathrm{Z_C} \end{bmatrix} + \mathrm{T} \end{gathered} \tag{4}$$

where $\mathrm{Z_C}$ is the depth in camera frame, directly given by the depth camera. $[\mathrm{X_C}\ \mathrm{Y_C}\ \mathrm{Z_C}]$ is the point in camera frame.

4. Method

4.1. Overview

The transformer-based visual model proposed in [20] demonstrates remarkable resilience to severe occlusion, disturbance, and displacement. Taking cues from VIT, we endeavored to leverage the transformer to enhance the contrast between global and local during grasp detection. Furthermore, we designed an efficient and intuitive way to link the transformer with grasping CNN for feature fusion.

An overview of the grasp model is depicted in Figure 2. We have designed a symmetric structure based on a transformer to effectively map features between parts and the whole in complex backgrounds. More specifically, the input, which is a 2D image, is divided into non-overlapping patches, resulting in a sequence of image-related vectors that act as tokens. These sequences act as the input of multiple attention layers, providing a more comprehensive analysis of the parts and the whole. Additionally, we reshaped the flattened output sequences into the raw size by using a trainable linear projection. These sequences are then concatenated in their original positions, bridging the gaps between the transformer and CNN, resulting in better multi-scale feature fusion. Finally, at the top of the model, the grasping CNN receives the output of the projection as input to predict potential grasps.

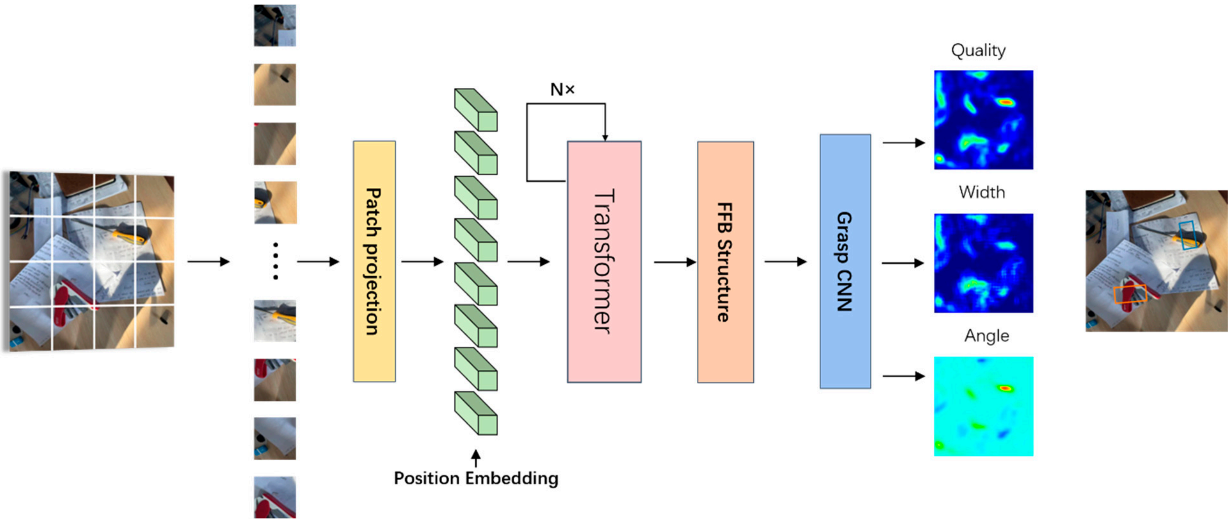

Figure 2. The architecture of our grasping detection mode.

4.2. Transformer-Based Feature Extraction

The standard transformer is strict about the size of the input, taking as input a 1D sequence of token embeddings. Before being fed into the transformer blocks, the input is evenly cut into flattened 2D patches of the image, and then each is resized into the 1D shape through a projection layer, which yields the vector sequences as the tokens for the contiguous transformer blocks. Specifically, a visual image $\mathrm{I} \in \mathbb{R}^{\mathrm{W}\times\mathrm{H}\times\mathrm{C}}$ is divided into equally sized patches $\mathrm{X} \in \mathbb{R}^{\mathrm{N}\times(\mathrm{P}^2\times\mathrm{C})}$, where (W,H) is the resolution of the original image, C is the number of channels, P represents the size of each patch, and $\mathrm{N} = \mathrm{H}\times\mathrm{W}/\mathrm{P}^2$ represents the number of total patches. The immediately following projection layer flattens and maps these image patches within D dimensions, an effective input sequence length $\mathrm{X} \in \mathbb{R}^{\mathrm{N}\times\mathrm{D}}$ for the transformer.

The attention mechanism in the transformer is a crucial component that improves the comparison and combination of local and global features. It has the ability to establish interactions across pixels, regardless of their spatial distance. The structure of the transformer block is presented in Figure 3.

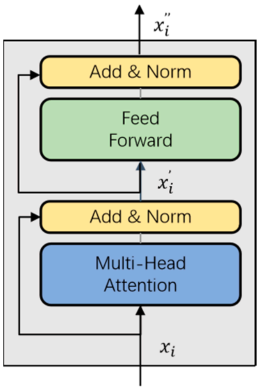

Figure 3. The transformer block.

In particular, we used multiple-head attention (MHA), which does not share the corresponding parameters, provides flexibility on different features, and reduces processing time due to parallel computing. The attention between image tokens is as follows:

$$\begin{gathered}\mathrm{Attention}(\mathrm{Q},\mathrm{K},\mathrm{V})=\mathrm{SoftMax}\left(\frac{\mathrm{QK^T}}{\sqrt{\mathrm{d}}}\right)\mathrm{V}\\ \mathrm{Q}=\mathrm{XW_Q^T},\ \mathrm{K}=\mathrm{XW_K^T},\ \mathrm{V}=\mathrm{XW_V^T},\end{gathered}\tag{5}$$

where the Q, K, and V vectors are obtained by multiplying the tokens $\mathrm{X}\in\mathbb{R}^{\mathrm{N}\times\mathrm{D}}$ by the corresponding weight matrix $\mathrm{W_i^T}\in\mathbb{R}^{\mathrm{D}\times\mathrm{D}}$, $\mathrm{W_i^T}$ is implemented by the fully connected layer, and d is a scale parameter.

For grasp detection in complex backgrounds, the transformer is very helpful with visual awareness. As with [14,15], we rigorously stacked the equal-sized transformer blocks to extract features from images of complex backgrounds. Accounting for the model runtime matching grasp detection in real-life settings, different sets of parameters were carefully designed to create a delicate balance between speed and accuracy without poor imitation; more details are described in the experiments section. The transformer-based feature extraction uses constant latent vector size D through all of its layers and ends up with the token vectors of the same size as the input $\mathrm{X}\in\mathbb{R}^{\mathrm{N}\times\mathrm{D}}$, with more holistic semantic information, through a forward propagation. The computation steps of the transformer block are represented as follows:

$$\begin{aligned}\mathrm{x_i'}&=\mathrm{LN}(\mathrm{MHA}(\mathrm{x_i})+\mathrm{x_i})\\ \mathrm{x_i''}&=\mathrm{LN}(\mathrm{FFN}(\mathrm{x_i'})+\mathrm{x_i'})\end{aligned}\tag{6}$$

where $\mathrm{x_i}$ denotes the output from the previous layer. LN refers to layer norm that normalizes each sample rather than a batch, and FFN is a simple fully connected network.

4.3. Feature Fusion Bridge

Previous works on the visual transformer separated the output patch tokens and directly utilized them for different detection heads, such as classification. However, this method is not suitable for grasp detection, particularly in complex backgrounds, as partial aggregation can lead to a loss of better information representation. We figured out a more efficient way to connect that ensures all parts of the model remain in order, while also focusing on the object's grasp.

In general, the classical transformer relies on flattened 1D sequences, while CNN requires at least 2D, such as an image. To handle that, firstly we used a fully connected layer to adjust the output sequence features of transformer blocks to the original patch token size $\left\{ \mathrm{D} \rightarrow (\mathrm{P}^2 \times \mathrm{C}) \right\}$, defined as:

$$\mathrm{f(x)} = \sum_{\mathrm{i}=1}^{\mathrm{d}} \mathrm{w_i x_i} + \mathrm{b} \tag{7}$$

where w represents the weight parameters, b represents the bias, and then the tokens $\mathrm{X}' \in \mathbb{R}^{\mathrm{N}\times(\mathrm{P}^2\times\mathrm{C})}$ are reshaped into small-scale patch feature $\mathrm{N} \times \mathrm{Y}, \mathrm{Y} \in \mathbb{R}^{\mathrm{P}^2\times\mathrm{C}}$, each of which is an integration of its own location and of the whole. We arranged these vectors according to the spatial position of the original patches to obtain a feature map with the same resolution as the original image $\mathrm{I}' \in \mathbb{R}^{\mathrm{W}\times\mathrm{H}\times\mathrm{C}}$, as shown in Figure 4.

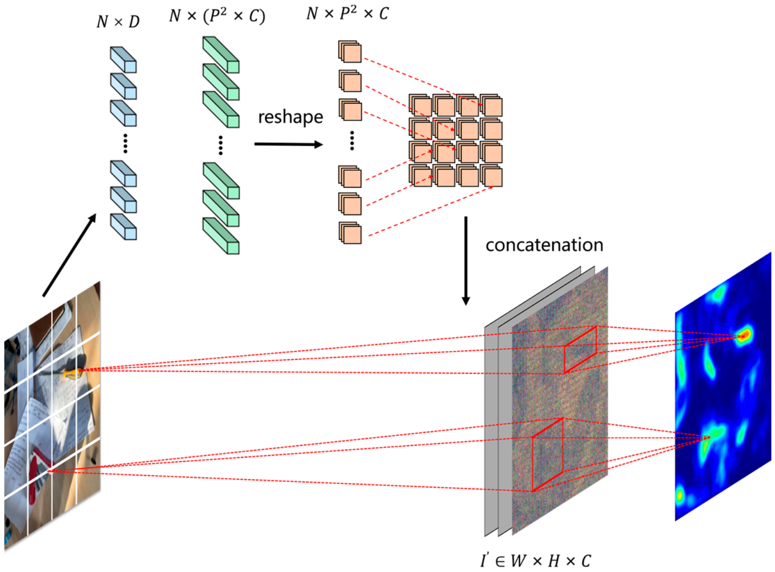

Figure 4. The feature fusion bridge.

An advantage of this approach is that all token features are aggregated at the same time in one forward propagation, preserving more global features, and another is that it is not affected by the input patch size of the transformer, because it always corresponds to the input, forming a symmetric structure.

4.4. Grasping CNN

We used GR-ConvNet [7] as the grasping CNN, which can directly output pixel-level grasping end-to-end without setting prior boxes, as shown in Figure 5. The network consists of down-sampling layers, residual layers, and up-sampling layers, forming a symmetric encoder–decoder structure. Four grasping detection heads are naturally integrated into the end of the network, generating pixel-wise grasp predictions, outputting the grasp quality feature map $\mathrm{Q} \in \mathbb{R}^{224\times224}$, the gripper angle feature maps including $\mathrm{Sin}2\theta \in \mathbb{R}^{224\times224}$ and $\mathrm{Cos}2\theta \in \mathbb{R}^{224\times224}$, and the gripper opening width feature map $\mathrm{W} \in \mathbb{R}^{224\times224}$.

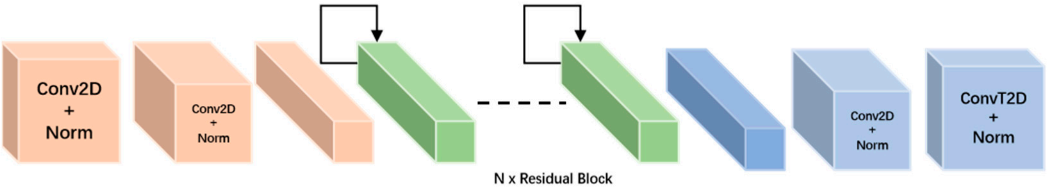

Figure 5. Diagram of GR-ConvNet.

The grasp quality map is composed of 0 to 1, representing the possibility of grasps at the corresponding pixel. The θ and W indicate the rotation angle and opening distance of the grasp, respectively.

4.5. Loss Function

Given the input image I, the model predicts a set of grasp pixel heatmaps $G_i = \{Q_i, Sin2\theta_i, Cos2\theta_i, W_i\}$, and the ground truth map $\widehat{G}_i = \left\{\widehat{Q}_i, \sin 2\widehat{\theta}_i, COS2\widehat{\theta}_i, \widehat{W}_i\right\}$ is set to the same format as the grasping CNN's outputs. We trained the model by calculating the $Smooth_{L1}$ loss between the prediction and labels, treating the grasp detection problem as a regression problem, which is beneficial for algorithm implementation. The loss function is defined as follows:

$$L\left(\widehat{G}_i, G_i\right) = \frac{1}{N}\sum_{i}^{N} \sum_{m \in \{q, \cos 2\theta, \sin 2\theta, w\}} Smooth_{L1}\left(\widehat{G}_i^m - G_i^m\right) \tag{8}$$

where $Smooth_{L1}$ is defined as

$$Smooth_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

5. Experiments

In this section, we conducted extensive experiments to validate the potential improvement of the suggested structure in grasp detection. We evaluated this on public grasp datasets such as Cornell [5] and VMRD [21]. We used different parameter sets to determine the optimal performance of the proposed structure on various datasets. We also explored different training strategies to comprehend the data requirements of each model. Moreover, we collected real-world images to demonstrate that the proposed FFB structure improves grasp detection performance, particularly when dealing with complex backgrounds.

5.1. Datasets and Implementation Details

The Cornell dataset [5], consisting of 885 images, each containing one grasping object from 24 different object categories, has been widely used for grasp detection. Due to the small size of the original dataset, we performed data augmentation, including random cropping, scaling, and rotation, to meet training requirements. Additionally, we conducted experiments on a bigger and more complex multi-object dataset, the VMRD [21], to validate the model. The VMRD, consisting of 5185 images containing 17,688 object instances from 31 object categories, provides 51,530 manipulation relationship labels in total. Some examples of the datasets are in Figure 6. In the experiment, all ground truth grasping boxes were used to involve training.

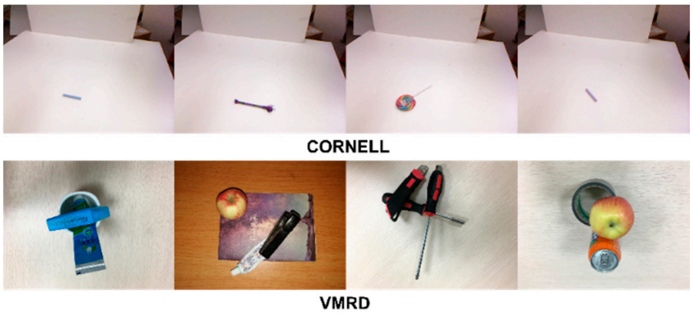

Figure 6. Instances in the datasets.

Two-Dimensional Gaussian-Based Grasp Augmentation: The dataset images were cropped, scaled, and normalized to match the input of the model. For the grasping quality ground truth, previous research [7] filled the grasp rectangle pixels with 0 and 1, assigning equal probabilities to the object edge and center. In [22], a Gaussian kernel was used to encode the grasp representation and highlight the object center. Furthermore, we adopted and expanded by using a 2D Gaussian kernel to adjust the grasp label at the pixel level in the grasp quality ground truth $\mathrm{I_q} \in \mathbb{R}^{224\times224}$. This method not only identifies the object's center position but also provides some orientation, as shown in Figure 7.

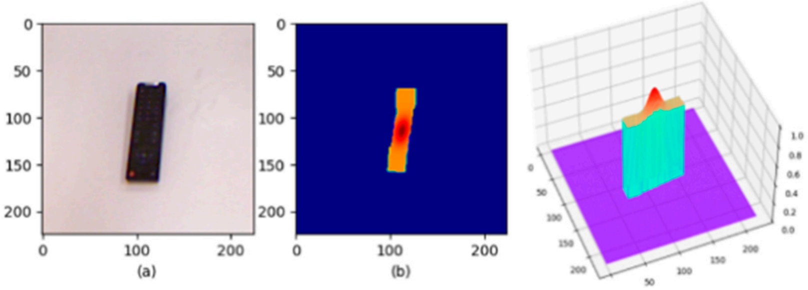

Figure 7. The grasp quality label map with 2D Gaussian kernel. Subfigure (**a**) is the original image, and subfigure (**b**) is its grasp quality label.

Specifically defined as:

$$\mathrm{Q(P_i)} = \frac{1}{(2\pi)^{\mathrm{d}/2}\left|\Sigma\right|^{1/2}} \exp\left[-\frac{1}{2}\left(\mathrm{P_i} - \mathrm{u_{I_q}}\right)^{\mathrm{T}}\Sigma_{\mathrm{I_q}}^{-1}\left(\mathrm{P_i} - \mathrm{u_{I_q}}\right)\right] \tag{9}$$

where

$$\mathrm{P_i} = (\mathrm{x_i}\,,\mathrm{y_i}) \ \in \mathrm{I_q}$$

where d = 2 represents the dimension of the vector; $\mathrm{u_{I_q}} = \{\mathrm{u_x}\,,\mathrm{u_y}\}$ represents $(\mathrm{x_i}\,,\mathrm{y_i})$ of the mean; Σ is the covariance matrix, describing the correlation of $(\mathrm{x_i}\,,\mathrm{y_i})$.

$$\Sigma = \begin{pmatrix} \delta_1 & \delta_2 \\ \delta_3 & \delta_4 \end{pmatrix} \tag{10}$$

The covariance matrix Σ is set empirically to conform to the orientation of objects in the data set as much as possible.

Training configuration: The entire grasp system was achieved using Pytorch 1.11.0 with CUDA 11.3 packages on Ubuntu 20.0. During the training period, the model was trained end-to-end on an AMD Ryzen 5 5600X CPU and an Nvidia RTX3080Ti GPU with 12 GB of memory.

Training schedule: We used a three-stage training strategy for the models. First, the grasping CNN of the model was trained on target datasets alone; the best-performing weights were used for the next stage of training. Second, we froze the CNN section and trained the rest sections including the transformer block and the FBB. Third, finally, we unfroze the grasp CNN and trained the entire model end-to-end. During the training process, the stochastic gradient descent (SGD) optimizer was used to optimize the model's backpropagation, with an initial learning rate of 0.001 and a learning rate decay of 0.1.

For the Cornell dataset, the batch size was set to 16, and the model was trained for a total of 200 epochs, with 50 epochs for the first stage, 50 epochs for the second stage, and 100 epochs for the final stage. For the VMRD, the batch size was set to 8, and the model was trained for a total of 500 epochs, which were split into 100, 150, and 250 epochs, respectively, for each stage. In each training phase, we periodically saved the weight of the model and tested it, and proceeded to the next phase of training when the performance had leveled off.

5.2. Model Variants

We carefully implemented the transformer parameter configurations, without relying on the basis, to determine the potential of the proposed model, as summarized in Table 1. The resolution of the transformer is affected by the input patch size. Smaller patch sizes will divide the original image more carefully but with a higher computational expense. In this study, we primarily set the input patch size of the transformer blocks to 7 × 7 to achieve more accurate grasp detection. Additionally, we scaled the models by adjusting the hidden size, MLP size, and depth to balance accuracy and computation on the Cornell dataset and VMRD.

Table 1. Details of vision transformer model variants.

Patch Size	Hidden Size	MLP Size	Layers	Params
7	256	512	3	4 M
7	256	512	6	8 M
7	256	512	9	12 M
7	512	1024	9	28 M
7	1024	2048	6	51 M
16	1024	2048	6	22 M

5.3. Metrics

Similarly to previous works [6,7,17], we adopted the same evaluation metrics to assess the performance of our model. Specifically, a predicted grasp was considered feasible when it satisfied the following conditions:

(1) The angle difference between the predicted rectangle and the annotated rectangle is less than 30°.

(2) The intersection over union (IOU) score between the predicted rectangle and the annotated rectangle is more than 25%.

$$\mathrm{J}(\mathrm{A},\mathrm{B}) = \frac{\mathrm{A}\cap\mathrm{B}}{\mathrm{A}\cup\mathrm{B}} \tag{11}$$

where A is the grasp rectangle label and B is the predicted grasp rectangle. When the overlapping areas of the two are similar and the direction is the same, it is considered to be a good grasp.

5.4. Experiments on Cornell Dataset

Following five-fold cross-validation as in previous works [7,16,22], we used the image-wise split method to test our model, where all images in the dataset were randomly sorted, and the average of five-fold cross-validation was the final result.

Results: We chose the best-performing of all parameter configurations, regardless of calculation cost. Table 2 shows the performance of our method compared to other works on the Cornell dataset. In the single-object dataset, our proposed model achieved a detection accuracy of 97.7% by taking RGB images as input, outperforming other algorithms. We used GR-ConvNet as the grasping CNN with an inference time of 2.8 ms. Compared to the algorithm in [16], the proposed transformer-based feature extraction structure only required 1.7 ms, and the total inference time of our model, 4.5 ms, was acceptable and met real-time requirements.

Table 2. Accuracy of different methods on Cornell grasp dataset.

Author	Method	Input	Accuracy (%)	Time (ms)
Jiang [4]	Fast Search	RGB-D	60.5	5000
Morrison [17]	GG-CNN	D	73.0	19
Lenz [5]	SAE	RGB	75.6	1350
Redmon [6]	AlexNet,	RGB-D	88.0	76
Zhou [23]	ResNet-101	RGB	97.7	117
Cao [22]	Efficient Grasp	RGB	95.3	6
Kumra * [7]	GR-ConvNet	RGB	96.6	2.8
Wang * [16]	TF-Grasp	RGB	96.7	3
Ours	Trans-CNN	RGB	97.7	4.5

The runtime results for the method * tested by ourselves; the other methods are referred to in the corresponding papers.

For the single-object grasping Cornell dataset, the highest score point in the quality map was set as the center of the grasping rectangle to obtain the corresponding width and angle. For the selected grasping pixel point, its index was recorded to find the width and angle of the corresponding position. Half of the width was set to the height of the grasp rectangle, and its corner points were easily solved with the angle value. The grasp detection results are visualized in the fourth row. The proposed method provides feasible grasps for objects with different shapes and positions, as shown in Figure 8.

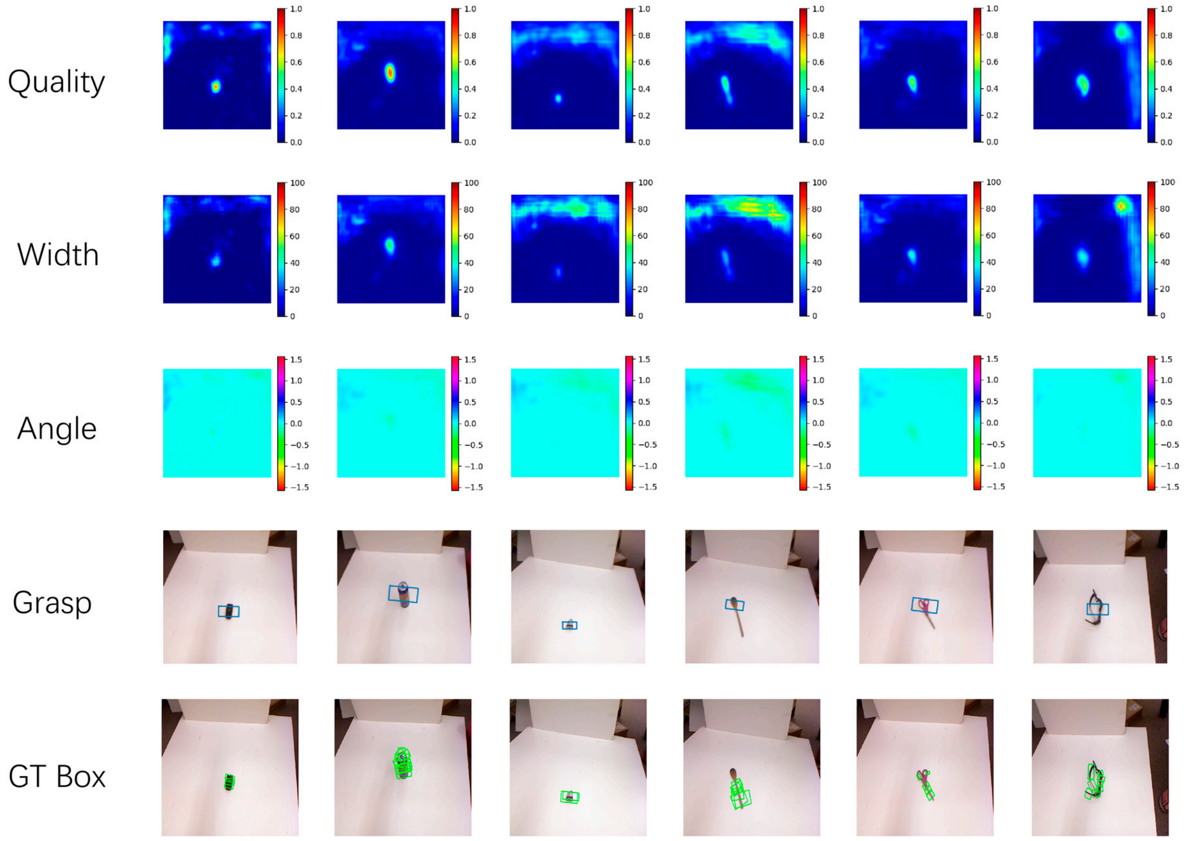

Figure 8. The detection results on the Cornell dataset.

5.5. Experiments on VMRD

We used the VMRD to test the performance of our model on multiple objects and stacked objects, which better aligns with real-world task requirements. As in [24–26], we employed the same dataset partitioning method provided by [21]. The images in the dataset were resized to 224 × 224 to be fed into the model, and we evaluated the model's performance in multi-object contexts by selecting the top three grasping candidates with the highest quality score. Meanwhile, we set the pixel distance threshold to avoid overlapping when selecting the most suitable pixel point.

Results: The comparison of our proposed method with other works on the VMRD dataset is shown in Table 3. In [24–26], grasp detection is defined as a two-stage method based on object detection, which complicates the model computation and increases inference time. We tested the one-stage detection method [7] on the VMRD dataset, and our proposed model achieved a 4.5% performance improvement compared to [7], demonstrating better detection accuracy. The grasp detection examples on the VMRD dataset are shown in Figure 9, and we drew the top three grasping boxes of each sample.

Table 3. Accuracy of different methods on the VMRD.

Author	Method	Accuracy (%)	Time (ms)
Zhang [24]	ROI-GD	68.2	110
Zhang [25]	ROI-ResNet	70.5	154
Park [26]	OD, GD, reasoning	74.3	30
Kumra * [7]	GR-ConvNet	67.7	2.8
Ours	Trans-CNN	72.2	5.2

The runtime results for the method * are tested by ourselves; the other methods are referred to in [26].

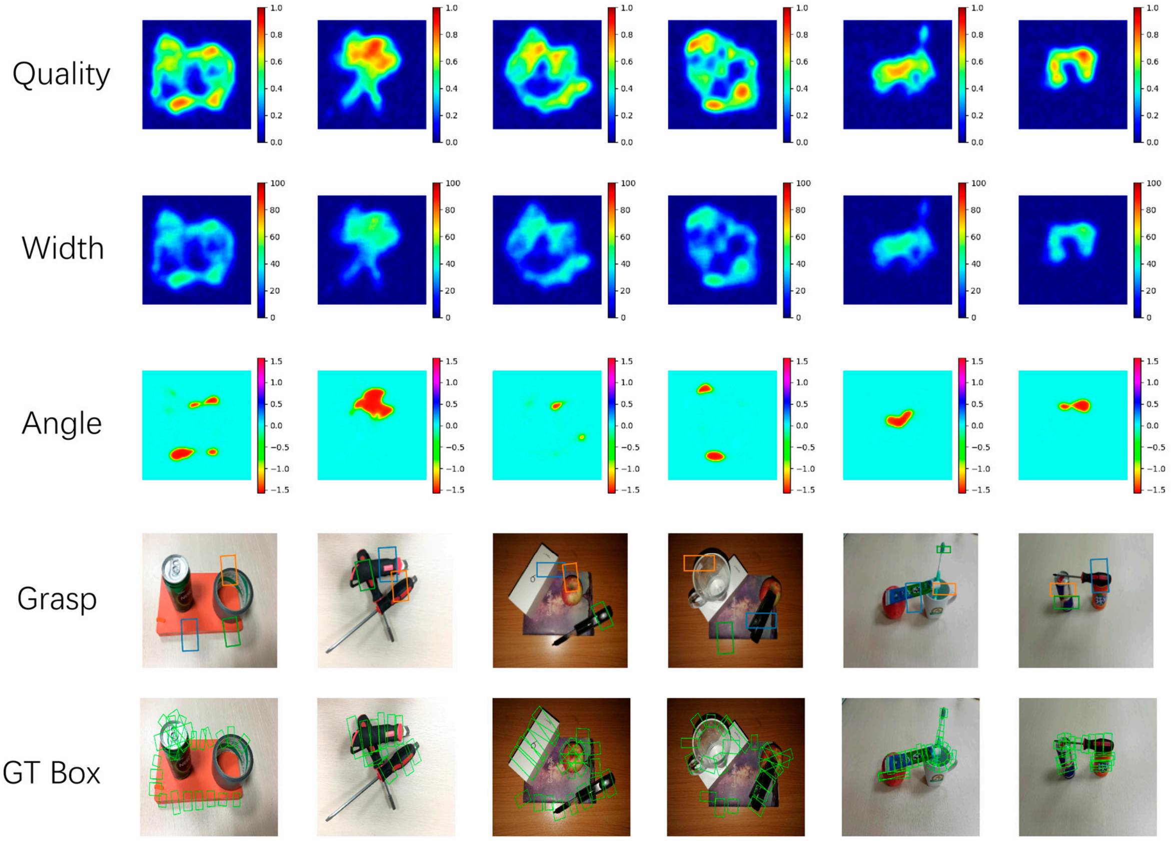

Figure 9. The detection results on VMRD.

5.6. Ablation Study

In Subsection B, the parameter configurations of the transformer layers in the model are described in detail. To further explore the impact of different configurations on grasp detection, we conducted experiments on the Cornell dataset and VMRD separately; the detailed experimental results are shown in Tables 4 and 5. The P, H, M, and L represent patch size, hidden size, MLP size, and layer, respectively.

Table 4. The accuracy of different configurations on the Cornell dataset.

P/H/M/L	Without 2D Gaussian Kernel	With 2D Gaussian Kernel
7/256/512/3	93.2%	94.3%
7/256/512/6	96.6%	97.7%
7/256/512/9	95.5%	95.5%
16/1024/2048/6	96.6%	96.6%
7/1024/2048/6	97.7%	97.7%

Table 5. The accuracy of different configurations on VMRD.

P/H/M/L	Without 2D Gaussian Kernel	With 2D Gaussian Kernel
7/512/1024/9	69.3%	70.2%
7/1024/2048/6	71.7%	72.2%

On Cornell: Five parameter configuration sets were tested on the Cornell dataset to verify the impact of different parameters on the proposed model. For a small dataset such as Cornell, the accuracy of the model was not improved significantly; with the growing parameters, the transformer-based feature extraction structure with small-scale parameter configuration can achieve excellent grasp detection accuracy.

On VMRD: For the larger and more complex dataset, the VMRD, the smaller models hardly fit well. Just stacking transformer layers singularly may not improve the performance further when the hidden size and MPL size are not large enough. The key to improving performance is to scale up the size, but that made the model bloated. We failed to achieve good results in the case of small size H and M (H = 256, M = 512). Instead, with increasing size, the proposed model achieved better detection accuracy, while the 2D Gaussian-based grasp representation also played a positive role.

5.7. Generalization Capability

The generalization capability of the model is critical in determining its practical applicability during grasp detection. To compare the generalization capability of the proposed model with that of CNN, we used the optimal weights of both models as grasping feature extractors on the Cornell dataset and tested detection accuracy on VMRD. Note that the models were trained only on the single-object Cornell dataset, and the multi-object VMRD was not part of the training phase. Figure 10 shows the generalization capability of different models for multi-object grasp detection. The models detected multiple grasping rectangles at once, and we selected the top N as the final result. As shown, even though both used the same Cornell dataset, our model showed better accuracy and generalization performance for multi-object grasp detection.

5.8. Grasp Detection in Complex Backgrounds

Our main objective was to enable the application of the grasp detection model in complex backgrounds. To evaluate the capabilities of our model in these scenarios, we captured real-life images with objects placed in complex backgrounds and assessed its performance using the model trained on the Cornell dataset, as shown in Figure 11. To help better understand how the proposed model makes grasping judgments about objects, we visualized the heatmap of the quality feature map, as show in the second and fourth rows.

The validity of the FFB: Is the proposed structure, which comprises transformer blocks and the bridge structure, effective for identifying prediction? To address this question, we visualized the output feature maps of the proposed structure to examine their impact during the prediction process. The visualizations are shown in the fifth row. It is apparent that the proposed structure's output demonstrates a high degree of similarity to the quality feature map, indicating a significant impact on the final detection results.

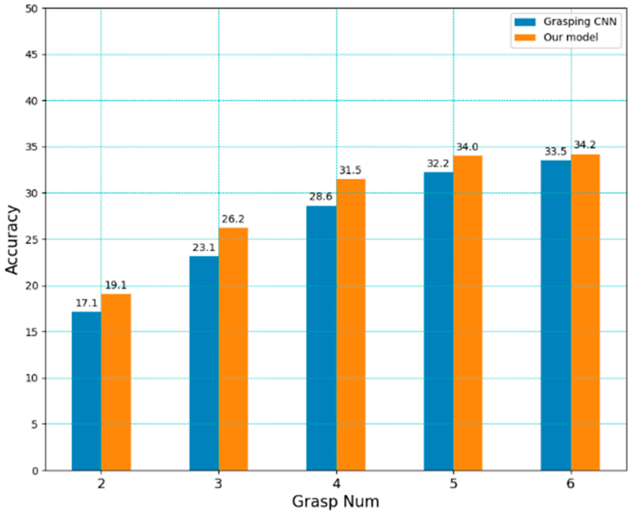

Figure 10. The accuracy of different models on multi-object VMRD.

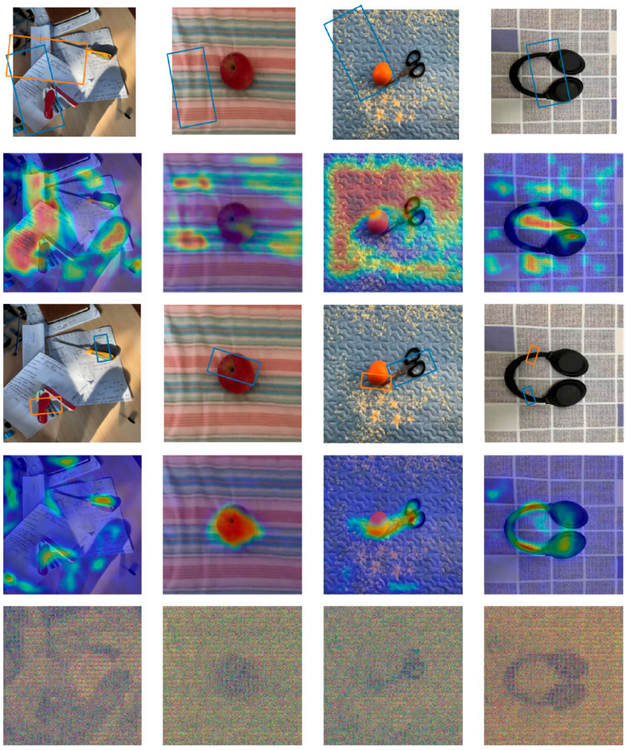

Figure 11. Examples of grasp detection by grasping CNN [7] and our method. The first and second rows show the results for CNN. The third, fourth and fifth rows show the results for our method.

Furthermore, we tested the grasp detection performance of the proposed method in more complex backgrounds. These scenes contained wide ranges of irregular patterns, or orderly arranged color patterns. These patterns are difficult to be mitigated by traditional image processing methods and cause strong interference for detection. The results are as shown in Figure 12.

For Scenes II and III, our model could easily detect suitable grasps. For Scenes I and IV, the more complex patterns in the background became an obstacle, and our model only responded positively to part of the objects. Even though it was trained on the simple single-object Cornell dataset, it showed advantages compared to the grasping CNN.

The experiments showed that our method can model the relationships between objects and the entire scene, as it induced and expressed the visual relationships between object features and background configurations. These designed structures positively impacted the grasping prediction and visually exhibited a segmentation-like effect.

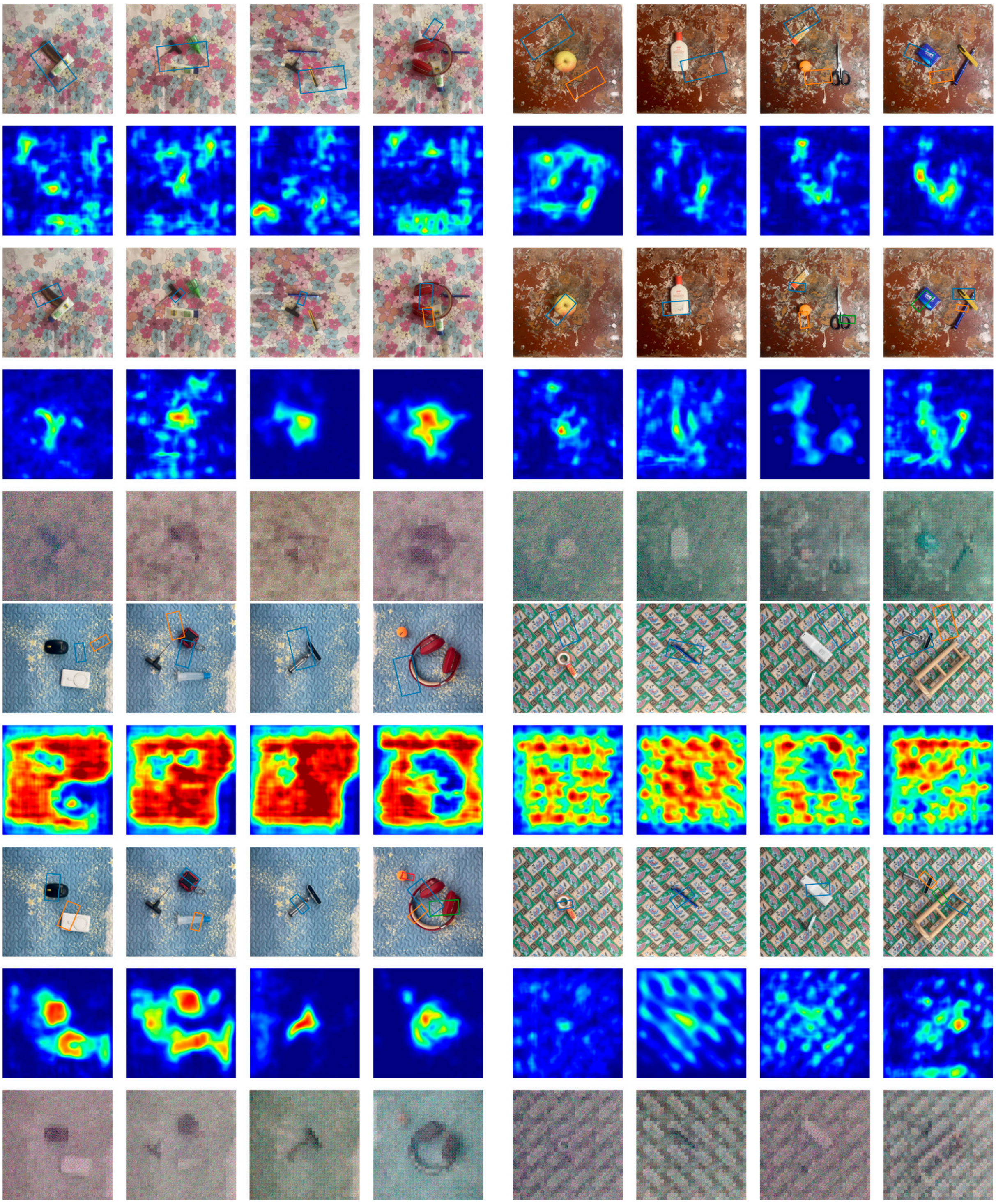

Figure 12. Examples of the results in more complex scenes. Scene I on the **upper left**, Scene II on the **upper right**, Scene III on the **lower left**, and Scene IV on the **lower right**.

5.9. Failure Case Discussion

In the experiments, a few inappropriate grasping poses could not be ignored. As shown in Figures 8 and 11, for irregularly shaped objects like scissors, the correct grip should be perpendicular to the edge. Our model failed to predict the correct grasping position and fell short for complex-shaped objects. Additionally, in Figure 12, it failed to detect small-scale objects in more complex scenes. This problem could be mitigated by adding objects with more-challenging shapes to the training data or using multimodal data.

6. Conclusions

In this paper, we introduced the transformer to improve the performance of CNN models for grasp detection in complex scenes. The proposed bridge between the transformer and CNN can achieve a reasonable feature fusion and a smooth transition. Experiments on public datasets (e.g., Cornell and VMRD) showed that the method surpasses the original grasp capacity of CNNs. Our method can isolate graspable objects from the background and achieve viable grasping poses in complex scene detection experiments. Future work will aim to incorporate multimodal information, along with self-attention mechanisms, to attain superior detection accuracy and more extensive grasps.

Author Contributions: Conceptualization, J.N. and S.L.; methodology, J.N. and S.L.; software, S.L. and H.L.; validation, S.L., H.L. and T.Z.; writing—original draft preparation, S.L., H.L. and T.Z.; writing—review and editing, L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Project of Songshan Laboratory under Grant (No. YYJC062022012), Key Scientific Research Projects of Colleges and Universities in Henan Province under Grant (No. 22A460022), Training Plan for Young Backbone Teachers in Colleges and Universities in Henan Province under Grant (No. 2021GGJS077), Zhongyuan Talent Program under Grant (No. ZYYCYU202012112), North China University of Water Resources and Electric Power Young Backbone Teacher Training Project under Grant (No. 2021-125-4), and North China University of Water Resources and Electric Power Master's Innovation Capacity Enhancement Project (No. NCWUYC-2023066).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Gruber, S. Robot hands and the mechanics of manipulation. *IEEE J. Robot. Autom.* **1987**, *75*, 1134. [CrossRef]
2. Bicchi, A.; Kumar, V. Robotic grasping and contact: A review. In Proceedings of the Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), San Francisco, CA, USA, 24–28 April 2000; Volume 1, pp. 348–353.
3. Schölkopf, B.; Platt, J.; Hofmann, T. Robotic Grasping of Novel Objects. In *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*; MIT Press: Cambridge, MA, USA, 2007; pp. 1209–1216.
4. Jiang, Y.; Moseson, S.; Saxena, A. Efficient grasping from RGBD images: Learning using a new rectangle representation. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3304–3311.
5. Lenz, I.; Lee, H.; Saxena, A. Deep learning for detecting robotic grasps. *Int. J. Robot. Res.* **2015**, *34*, 705–724. [CrossRef]
6. Redmon, J.; Angelova, A. Real-time grasp detection using convolutional neural networks. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 1316–1322.
7. Kumra, S.; Joshi, S.; Sahin, F. Antipodal Robotic Grasping using Generative Residual Convolutional Neural Network. 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. 9626–9633.
8. Pokorny, F.T.; Bekiroglu, Y.; Kragic, D. Grasp moduli spaces and spherical harmonics. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 389–396.
9. Saxena, A.; Wong, L.L.S.; Ng, A.Y. Learning grasp strategies with partial shape information. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, IL, USA, 13–17 July 2008; Volume 3, pp. 1491–1494.
10. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]

11. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
12. Karen, S.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
13. Kumra, S.; Kanan, C. Robotic grasp detection using deep convolutional neural networks. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 769–776.
14. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Trans-formers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
15. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 6000–6010.
16. Wang, S.; Zhou, Z.; Kan, Z. When Transformer Meets Robotic Grasping: Exploits Context for Efficient Grasp Detection. *IEEE Robot. Autom. Lett.* **2022**, *7*, 8170–8177. [CrossRef]
17. Douglas, M.; Corke, P.; Leitner, J. Learning robust, real-time, reactive robotic grasping. *Int. J. Robot. Res.* **2019**, *39*, 183–201.
18. Zhang, Z. Flexible camera calibration by viewing a plane from unknown orientations. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 1, pp. 666–673.
19. Horaud, R.; Dornaika, F. Hand-Eye Calibration. *Int. J. Robot. Res.* **1995**, *14*, 195–210. [CrossRef]
20. Naseer, M.; Ranasinghe, K.; Khan, S.; Hayat, M.; Khan, F.S.; Yang, M.-H. Intriguing Properties of Vision Transformers. *arXiv* **2021**, arXiv:2105.10497.
21. Zhang, H.; Lan, X.; Zhou, X.; Tian, Z.; Zhang, Y.; Zheng, N. Visual Manipulation Relationship Network for Autonomous Robotics. In Proceedings of the 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), Beijing, China, 6–9 November 2018; pp. 118–125.
22. Cao, H.; Chen, G.; Li, Z.; Feng, Q.; Lin, J.; Knoll, A. Efficient Grasp Detection Network with Gaussian-Based Grasp Representation for Robotic Manipulation. *IEEE/ASME Trans. Mechatron.* **2023**, *28*, 1384–1394. [CrossRef]
23. Zhou, X.; Lan, X.; Zhang, H.; Tian, Z.; Zhang, Y.; Zheng, N. Fully Convolutional Grasp Detection Network with Oriented Anchor Box. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 7223–7230.
24. Zhang, H.; Lan, X.; Bai, S.; Zhou, X.; Tian, Z.; Zheng, N. ROI-based Robotic Grasp Detection for Object Overlapping Scenes. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 4768–4775.
25. Zhang, H.; Lan, X.; Bai, S.; Wan, L.; Yang, C.; Zheng, N. A Multi-task Convolutional Neural Network for Autonomous Robotic Grasping in Object Stacking Scenes. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 6435–6442.
26. Park, D.; Seo, Y.; Shin, D.; Choi, J.; Chun, S.Y. A Single Multi-Task Deep Neural Network with Post-Processing for Object Detection with Reasoning and Robotic Grasp Detection. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 7300–7306.

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.