Appendix A: Method

A.1. Overlap Region Calculation

RoI Align converts the overlap region into a fixed-size feature map ($14 \times 14 \times 256$), and then the multi-branch feature combination module and overlap region transformer module proposed in this paper process the overlap region feature map to obtain a more fine-grained relationship identification. Specifically, we tried different size feature maps (e.g., $7 \times 7 \times 256$, $14 \times 14 \times 256$, $28 \times 28 \times 256$, etc.) and found that $14 \times 14 \times 256$ is the best choice considering the overall performance of the model.

A.2. Multi-Branch Feature Combination Module

The overlap region of the two objects is first transformed into a $14 \times 14 \times 256$ feature map through RoI Align. Then, the multi-branch feature combination module processes the $14 \times 14 \times 256$ feature map using convolution, deconvolution, and multi-branch dilation convolution to obtain three $28 \times 28 \times 256$ feature maps. Next, the $28 \times 28 \times 256$ feature maps of three different dilation convolution branches are concatenated to obtain a $28 \times 28 \times 768$ feature map. Finally, a series of convolutions are used to convert the $28 \times 28 \times 768$ feature map to $7 \times 7 \times 256$ and combined with the $7 \times 7 \times 256$ feature map obtained by the original overlap region through RoI Align.

In the detailed illustration of the multi-branch feature combination module. "3×3 conv" represents the convolution with kernel_size of 3, stride of 1, and padding of 1. "two 3×3 conv" represents two layers of "3×3 conv." "2×2 deconv" represents deconvolution with kernel_size of 2 and stride of 2. "5×5 conv" means convolution with kernel_size of 5, stride of 1, and padding of 2. "3×3 conv rate=1" is equivalent to "3×3 conv." "3×3 conv rate=2" represents a dilation convolution with kernel_size of 3, stride of 1, padding of 2, and dilation of 2. "2×2 conv" indicates a convolution with kernel_size of 2 and stride of 2.

A.3. Overlap Region Transformer Module

We propose the overlap region transformer module to use a vision transformer to obtain the global visual features of the overlap regions of two objects. RoI Align first converts the overlap region into a $14 \times 14 \times 256$ feature map and then uses the overlap region transformer module to obtain the self-attention of the overlap region. The $14 \times 14 \times 256$ overlap region feature is first flattened into 196×256. Then a 1×256 class token representing the global visual features of the overlap region is added, and the 197×256 position information corresponding to each token is added. And then, the 197×256 features are input to the encoder block to obtain the global visual features of the overlap region. Finally, obtain the 1×256 global visual features of the overlap region, which is the input to identify the fine-grained relationship between objects.

Standard self-attention is a popular building block for neural architectures. In the standard self-attention layer, the input vectors is first transformed into three different vectors, i.e., query vector q, key vector k and value vector v. The three vectors have the same dimension, i.e., $d_q = d_k = d_v = d_{model}$, where d_{model} is the dimension of the input vector. Vectors derived from three different inputs are then packed into three different matrices, namely, Q, K, and V. The attention function between different input vectors is calculated as follows:

- Compute scores between different input vectors with $S = QK^T$.
- Normalize the scores for the stability of gradient with $S_n = S/\sqrt{d_{model}}$.
- Translate the scores into probabilities with softmax function $P = softmax(S_n)$.
- Obtain the weighted value matrix with $W = PV$.

The process can be unified into one function:

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_{model}}})V. \quad (1)$$

Multi-head attention is an extension of standard self-attention. Multi-head attention runs h self-attention operations, called h heads, in parallel and projects their concatenated outputs.