TGA: Two-level Group Attention for Assembly State Detection

Hangfan Liu [*, †] TU Kaiserslautern DFKI[**]
Yongzhi Su [*, ‡] TU Kaiserslautern
Jason Rambach [§] DFKI[**]
Alain Pagani [¶] DFKI[**]
Didier Stricker [‖] TU Kaiserslautern DFKI[**]

Abstract

Assembly state detection, i.e., object state detection, has a critical meaning in computer vision tasks, especially in AR assisted assembly. Unlike other object detection problems, the visual difference between different object states can be subtle. For the better learning of such subtle appearance difference, we proposed a two-level group attention module (TGA), which consists of inter-group attention and intro-group attention. The relationship between feature groups as well as the representation within each feature group is simultaneously enhanced. We embedded the proposed TGA module in a popular object detector and evaluated it on two new datasets related to object state estimation. The result shows that our proposed attention module outperforms the baseline attention module.

Index Terms: Computing methodologies—Machine learning—Machine learning approaches—Neural networks; Computing methodologies—Artificial intelligence—Computer vision—Computer vision tasks; Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Mixed/augmented reality;

1 Introduction

As a user-friendly human-machine interaction technology, AR has advanced a lot in the last decades. It can be used in education [19], healthcare [29], manufacturing [22], etc. AR assisted assembly as one possible use case has also been researched a lot recently [26, 31, 32]. Objects which consist of several removable and adjustable components can have different states due to the assembly step and the assembly method. The object state detection task is one of the critical components in AR assisted assembly and automatic robotic manipulation.

For object state estimation, one possible way is to estimate 6 DoF pose of each component using pose estimation methods [8, 20]. Based on the relative pose of each component, the assembly state can be estimated. However, the pose estimation methods perform less accurately when the target objects are more or less occluded. Nevertheless, the components are usually occluded after the assembly. Another possible approach is to treat state estimation as a pure classification problem, which focuses on the whole object rather than each component.

However, unlike the typical image detection and classification problem, in our task, half-assembled products (see Fig. 1 top-left) and similar object states(see Fig. 1 second row) are usually occurred in the assembly process, which are a considerable challenge for the classification. Thus, we require a network that can learn the global structure and subtle visual differences between assembly states.

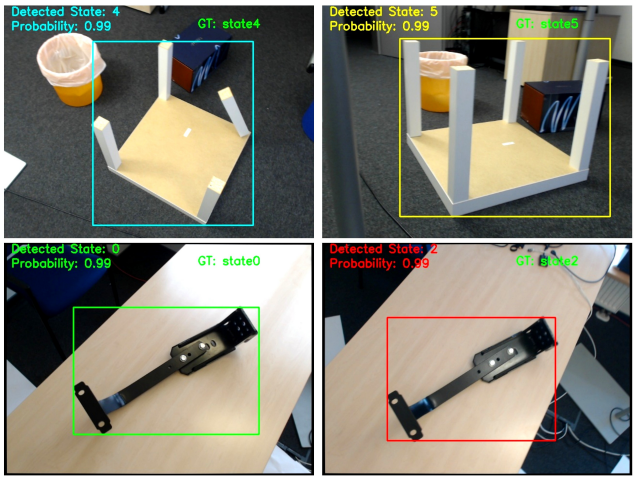

Figure 1: Object state detection has a critical meaning in computer vision tasks, especially in AR assisted assembly. The half-assembled assembly state have to be distinguished with finished assembly state (top images). The appearance similar object states (bottom images) also have to be recognized individually. In this work, we proposed a new attention module, i.e. TGA to solving this problem.

The fine-grained classification problem is similar to our problem since sub-class classification is performed. The visual difference between sub-classes is conceivably smaller than the difference between classes. Attention-based CNN [35] has achieved significant performance on fine-grained visual categorization task. They try to solve the sub-class classification by learning part detectors, part annotation, and cropping the detail parts with predefined quantity (e.g., the head, the body and the foot of the bird). However, their attention methods have several issues if applied to assembly task such as the one we are addressing. Specifically, the part annotation information will limit the robustness and generalization ability of the model. The number of required annotations varies for different objects. Moreover, the occlusion in the assembly makes the precise annotation more difficult.

To remedy the above problems, we proposed a two-level group attention module inspired by [15], which divides the feature maps into several groups and improves the semantic response in each feature group. We proposed a lightweight attention module named two-level group attention (TGA). It can strongly capture the detail feature from lots of subtle areas by 1) modelling the cross-group relationship (inter-group attention), and 2) enhancing the feature representation for each group (intro-group attention). Specifically, the former learns the global structure with the important factor between groups. The later captures features for a specific detail in each group. Furthermore, an assembling object usually consists of several different size components. To make the parts of different scales have the same representational power, we added a multi-branch trident

[*]equal contribution
[†]e-mail: Hangfan.Liu@dfki.de
[‡]e-mail: Yongzhi.Su@dfki.de
[§]e-mail: Jason.Rambach@dfki.de
[¶]e-mail: Alain.Pagani@dfki.de
[‖]e-mail: Didier.Stricker@dfki.de
[**]German Research Center for Artificial Intelligence (DFKI)

block [16] into the CNN backbone to select the appropriate receptive field by dilated convolutional layers. Collecting and labelling a massive dataset from the real-world is a cumbersome process. Thus we used the CAD models of the objects in different states to create synthetic training data without any manual annotation cost. We apply the domain randomization and the domain adaptation technique in the synthetic datasets to close the reality gap between the simulator and the physical world and make sure the model generalizes well in different real environments. The overall structure of the network is represented in Fig. 3.

We tested our TGA on two different datasets (see Fig. 1). The CNN is able to distinguish the difference between the half-assembled assembly state and finished assembly state in the IKEA-table dataset: The top-left image has been classified as state 3 while the top-right as state 4. And the CNN can also recognize the appearance similar object states in the Fender dataset (the Fender has been assembled in different length in the bottom images): bottom-left image has been classified as state 0 while the bottom-right as state 2.

Our contributions are summarized as follows:

- We apply bilinear pooling to obtain the relationship among feature groups, which builds the inter-group attention. It contains only 2 learn-able parameters, can be embedded in CNNs without extra complex computations.
- We proposed a two-level attention module (TGA). The first level is our proposed inter-group attention. The second level is intro-group attention. This design simultaneously enhances the cross-groups relationship as well as the representation within a group. The TGA can be easily embedded in the most CNN backbones.
- We embedded TGA in the backbone of an object detector and tested different CNNs with our object state datasets. The results show that the CNN with the proposed TGA outperforms other CNNs.
- The backbone with TGA can be used in many AR applications, e.g. [26] to build an AR assisted Assembly

The paper is organized as follows: We first give an overview of the related work in Sec. 2. The attention module and overall architecture are described in detail in Sec. 3. Subsequently, we present our dataset used for training as well as testing in Sec. 4. Finally, a quantitative evaluation of our proposed TGA is presented in Sec. 5.

2 Related Work

2.1 Object Detection

Although anchor-free object detectors like FCOS [27] have rapidly developed in recent years, anchor-based object detectors are still the most popular one. The anchor-based object detectors can be divided into two categories, i.e., one-stage detectors, and two-stage detectors. One-stage detectors like YOLO family [2,23] and SSD [17] treat object detection as a regression problem. They take an input image and predefined anchors to predict the class probabilities and bounding box coordinates. One-stage detectors achieve high inference speed, whereas two-stage detectors are superior on location and recognition accuracy. The two-stage detectors like Faster R-CNN [24] and Mask R-CNN [6] consist of a coarse prediction stage and a finer prediction stage. In the first stage, adopt a Region Proposal Network (RPN) to generate regions of interest (ROIs). In the second stage, the proposed ROIs are sent down the pipeline for finer object classification and bounding-box regression.

At the same time, the scales of object instances could vary in a wide range. This scale variation makes the object detection problem more difficult. The benefit of the feature pyramid SSD [17] and MS-CNN [3] assign proposals to appropriate feature levels for inference. Instead, SNIP [14] creates an image pyramid to train objects on different image scales selectively. HyperNet [12] concatenates features from different levels to generate appropriate region proposals that could alleviate scale variation. Furthermore, TridentNet [16] constructs a parallel multi-branch feature architecture with different receptive fields and achieve significant improvements. We use the idea behind TridentNet in this paper to build our object detector.

2.2 Group Attention Mechanism

Lots of research [11, 18] published that the attention mechanism plays a significant role in human perception. Recently, attention mechanism has proven to be a potential means to enhance deep CNNs. In [9] an attention module was introduced to exploit the channel-wise relationship. CBAM [30] further considered both spatial-wise and channel-wise relationship. TASN [35] proposed a trilinear attention module that conducts bilinear pooling to obtain the relationship among feature channels.

The idea of group attention algorithm is from group convolution and feature clustering functions. The earliest idea of group CNN began with AlexNet [13] that divides features into two groups on different GPUs to save computing resources. Besides, there have been several attempts [4,34] to incorporate group convolutions processing to improve the performance of CNNs. Inspired by group convolution, ResNest [33] generalized the channel-wise attention [9] into feature-map group representation. SGE [15] proposed a spatial-wise attention mechanism inside each feature group, by scaling the feature vectors over all the locations with an attention mask to enhance the semantic entity representations within each feature groups.

Compared with our work, our attention mechanism can simultaneously learn 1) intra-relationships of channels within a group, and 2) inter-relationships between different groups. Similar to SGE, each feature group in the TGA has its own semantic representation, which may be a key detail of the assembling component, or it may be unimportant noise information.

3 Two-Level Group Attention Module (TGA)

In this section, we introduce the proposed TGA module and the overall network. We first describe our proposed inter-group attention module, which aims to learn the importance of relationships across feature groups. We further give a brief overview of the intro-group attention module (SGE [15]). Then we present the TGA module by series connecting the inter-group attention and the intro-group attention. In the end, we describe the entire CNN architecture.

3.1 Inter-Group Attention Module

Our Inter-group Attention Module is an attention-based computational unit consisting of 1) dividing the feature maps into groups and 2) learning importance factors for each group. The first row of Fig. 2 depicts an overview of our inter-group attention.

As the assumption of [15, 25], we expect that each feature group can gradually capture the semantic response of the detail discriminate part. Thus we divide the original feature map $\mathbf{F}_o \in \mathbb{R}^{c\times h\times w}$ into $\mathbf{F} \in \mathbb{R}^{g\times c'\times h\times w}$, with $c = g \times c'$. Then we apply average pooling layer cross g dimension to aggregate the representation of each group features. The feature map with aggregated group features has a shape of $g \times 1 \times h \times w$. After merging the first two dimensions, we denote it as $\mathbf{F}_G \in \mathbb{R}^{g\times h\times w}$.

As shown in [15], each group of the $\mathscr{F}_G$ corresponds to a special visual pattern. Inspired by trilinear attention [35], which conducts bilinear pooling to capture the relationship among feature channels. We use the same idea to obtain the relationship among groups according to the feature representation of each $\mathbf{F}_G$. We reshape the $\mathbf{F}_G$ to $\hat{\mathbf{F}}_G \in \mathbb{R}^{g\times hw}$, then the calculation of the attention mask $\mathbf{M}$ can be formulated as:

$$\mathbf{M} = N(N(\hat{\mathbf{F}}_G)\hat{\mathbf{F}}_G^{\mathbf{T}}) \ , \tag{1}$$

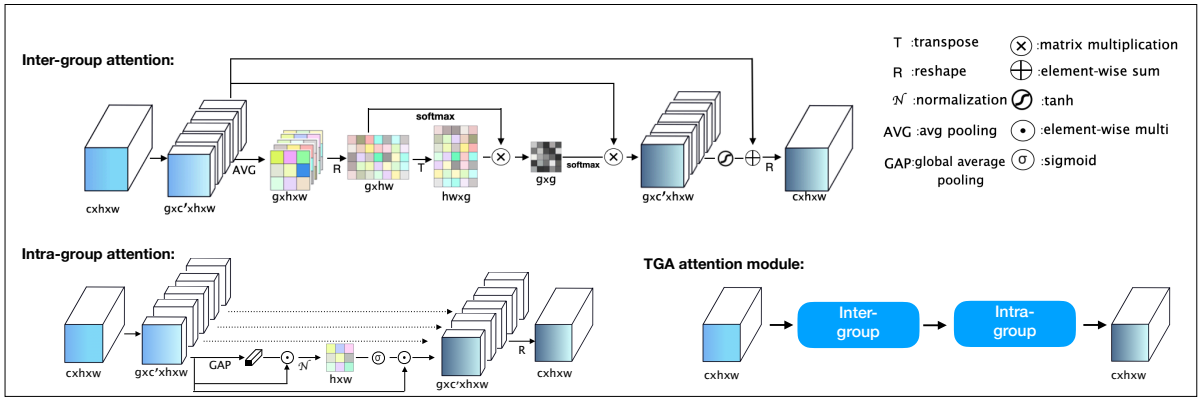

Figure 2: Illustration of our proposed TGA attention module. 1) inter-group attention module: the feature channels of each group is aggregated by average operation, then the bilinear pooling operation is used to obtain the attention guidance for each group. 2) intra-group attention module: we apply the idea of the SGE attention module proposed in [15], that is able to enhance semantic feature representation in each group.

where $N(\cdot)$ denotes softmax operation and $\hat{\mathbf{F}}_G^{\mathbf{T}} \in \mathbb{R}^{hw \times g}$ denotes the transpose of $\hat{\mathbf{F}}_G$. Subsequently, we obtain the attention mask among groups $\mathbf{M} \in \mathbb{R}^{g \times g}$. The inter-group relationship matrix $\mathbf{M}_{i,j}$ indicates the i_{th} group's influence on j_{th} group. We reshape the $\mathbf{F} \in \mathbb{R}^{g \times c' \times h \times w}$ into $\bar{\mathbf{F}} \in \mathbb{R}^{g \times c'hw}$, then apply matrix dot production over $\mathbf{M}$ and $\bar{\mathbf{F}}$ to obtain a matrix in $g \times c'hw$. The overall inter-group attention module can be summarized as:

$$\mathbf{F}' = reshape(tanh(\lambda \cdot (\mathbf{M}\bar{\mathbf{F}}))) \,, \tag{2}$$

$$\mathbf{F}'' = \mathbf{F}' \oplus \mathbf{F} \,, \tag{3}$$

where $\oplus$ denotes element-wise summation. The $\mathbf{M}\bar{\mathbf{F}}$ is scaled by a learn-able parameter λ, then take a tanh as activation function, and reshape operation to obtain the $\mathbf{F}' \in \mathbb{R}^{g \times c' \times h \times w}$. Finally reshape the output $\mathbf{F}''$ into the size of $c \times h \times w$ as the feature map applied with our inter-group attention.

3.2 Intra-Group Attention Module

As introduced above, we designed the entire TGA attention module structure with inter-group attention and intra-group attention. In our method, we use the SGE attention module proposed in [15] as our intra-group attention, which enhances the representation inside each feature group. The inter-group attention we proposed aims to emphasize or suppress each individual feature group. The intra-group attention can be summarized as:

$$\boldsymbol{sv} = \mathbf{F}_{gp}(\mathbf{X}) \tag{4}$$

$$\boldsymbol{coeff} = \gamma \cdot N(\boldsymbol{sv} \cdot \mathbf{X}) + \beta \tag{5}$$

$$\hat{\mathbf{X}} = \mathbf{X} \cdot \sigma(\boldsymbol{coeff}) \tag{6}$$

which $\mathbf{X}$ denotes a group of feature, and $\boldsymbol{sv}$ is the global average pooling of this group $\mathbf{X}$, indicates the semantic vector. The $\boldsymbol{coeff}$ denotes importance coefficients, the operation $N(\cdot)$ is normalization over the space dimension. This module involved only 2 parameters for each group: γ and β. Finally, to obtain enhanced group feature $\hat{\mathbf{X}}$, the original $\mathbf{X}$ is scaled by the generated importance coefficients $coeff$ via a sigmoid function gate $\sigma(\cdot)$ over the space. The intra-group attention module can be illustrated in the bottom left of Fig. 2.

3.3 Overall CNN Architecture

In our attention module, we adopt a sequential inter-intra arranging method. Furthermore, our TGA module is a lightweight attention unit without extra complex calculations (see bottom right of Fig. 2).

We build our object state detector based on Faster RCNN [24] with trident block [16]. The architecture of the entire CNN is depicted in Fig. 3. We embedded our TGA-module in the backbone Resnet50 [7] backbone, by placing our TGA-module after the last BatchNorm [10] layer of residual block on stage1 and stage2. A multi-branch trident block is used to extract high-level feature from the outputs by attention module. The trident block contains multiple dilated convolutions with various dilation rates. Each dilated convolution independently extracts the feature map. The output of backbone is sent to RPN Network for the proposal of ROIs. The ROIs will be aligned as fixed-size feature maps, which are further processed for the object states classification their bounding box regression. In the following experiment section, we will present our proposed method's performance on the object state detection task.

4 Dataset

Considering two different scenarios in AR assisted assembly. 1)The object can be assembled as multiple possible states. This is a common industrial assembly use-case. Typically, the states have a very similar appearance. After assembly, AR can be used to ensure the assembled state is the required state. 2)In the AR guided assembly [26], the system must be able to distinguish, if an assembly step is already finished, in order to show the next assembly step.

We captured two datasets 1)Fender-assembly and 2)IKEA-table, to test our proposed method in the two different scenarios, respectively. Table 1 shows statistics of these datasets.

Dataset		State 0	State 1	State 2	State 3	State 4	State 5	SUM
IKEA-table	real	161	180	140	147	122	21	771
	synthetic	25.2k	25.2k	25.2k	25.2k	25.2k	25.2k	151.2k
Fender-assembly	real	106	116	102	108	106	110	648
	synthetic	19.44k	19.44k	19.44k	19.44k	19.44k	19.44k	116.64k

Table 1: Statistics of these datasets used in our experiments.

4.1 Real Test Dataset

The industrial Fender-assembly dataset consists of 6 possible states. The Fender can be assembled in 3 different lengths. For each length, it can be assembled in 2 directions (6 state = 3 length * 2 direction). The appearance variance of different states is very small. However,

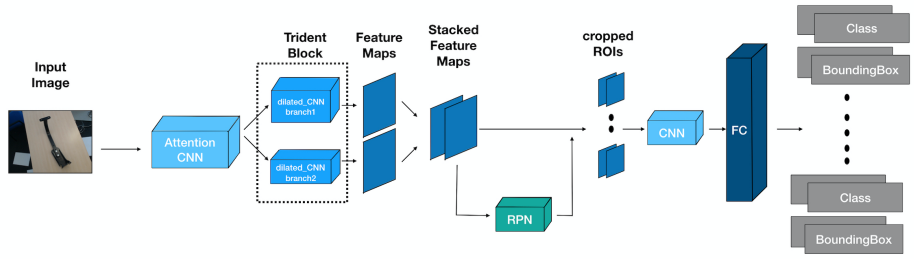

Figure 3: Overview of the entire CNN used for object state detection: Faster-RCNN detector embedded with TGA module and multi-branch trident block. The details of the network are summarized in the Sec 5.1.

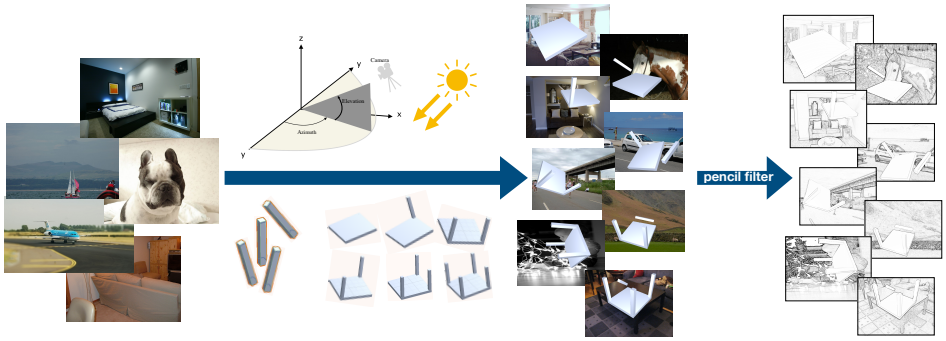

Figure 4: Synthetic objects with different assembly state are rendered on top of a random background (left), with random pose, random illumination, random light position, viewpoint and noise. For each state (except the final state), the distractor will randomly appear around its assembly location in the scene. Finally, all images will pass through pencil filter [21] as the input of the network

according to different product requirements on the production line, only one of the six states is correct.

The IKEA-table dataset also consists of 6 total possible states separately indicating each step of the manual assembly process. It is necessary to distinguish between the finished assembly and the half-assembled assembly, so that the AR system can show the correct guide for the assembly step. For each state i, the images involve the state i and the half-assembled state $i+1$ (since the half-assembled state $i+1$ is still not the state $i+1$, it will be labeled as state i).

The details of the assembly state label can be found in the corresponding object state graph (is illustrated in Fig. 5), which depicts the step-by-step assembly instructions.

4.2 Synthetic Training Dataset

Collecting and manually annotating a large number of training datasets is typically an expensive and time-consuming task in the real world. But it is easy to generate a synthetic dataset using a rendering engine.

In our work, we generated synthetic datasets with Unity3D using industrial CAD models. The 3D model of an object state is placed in a 3D scene with a random pose. The virtual camera rotates around the object with a step of 5 ± 2.5 degrees. In each viewpoint, the camera will translate to a random position within a pre-defined range to render the 3D object. Meanwhile, ground truth labels are automatically generated: state label and bounding box of objects.

In particular, the IKEA real test dataset contains many images, which depict an intermediate state (half-assembled state). Therefore when we generated the IKEA synthetic training dataset, besides the object state model, we added an additional table's leg model into the scene. The object state model, together with an unassembled table leg model, is used to simulate such half-assembled states. We call the unassembled table leg (or any other unassembled component) 'flying distractor'. 50% of the total images include the 'flying

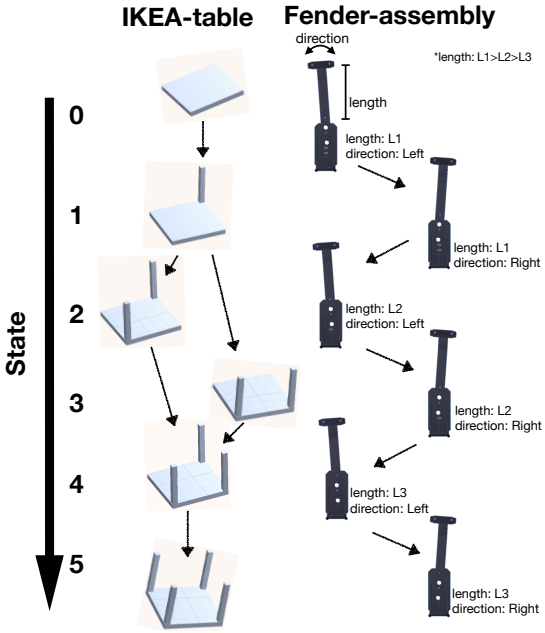

Figure 5: Assembly instruction step-by-step.

distractor'. We randomize the translation of the flying distractor obeying a normal distribution. Empirically, we define that the offset of the flying distractor along each translation axis obeys the normal distribution $N(0, 12.975)$. This distribution indicates that the offset has 30% chance to be between -5cm to 5cm. Note that the table size is measured in 50×50 cm. For a better training of the CNN, the image of half-assembled state should be adequate distinguished with

an image of fully assembled state). So in the training dataset, if the distractor' distance to the assembly location is less than 5cm, then its angular distance related to the assembly pose have to be limited to at least 15 degrees.

Considering the gap from synthetic to real-world data, we applied the technique of domain randomization [28], sampling the 3D model by randomly varying the several aspects of the scene:

- Location and angle of the virtual camera with respect to the scene(azimuth from 0° to 360° , elevation from 25° to 90°)
- Location of the point light
- Random background images from VOC2014 [5]
- Random illumination
- Random noise

We apply pencil filter as our pre-processing step for input images for both training and testing, which was successfully used in [21,26]. This pre-processing step forced the image to focus only on the edge information. Our models are trained on pencil images of the synthetic training images and evaluated on the pencil images of real test images. Taken Ikea-table dataset as an example, we show the whole process of data generation in the Fig. 4. The process suits arbitrary objects.

5 Experiments and Evaluation

In this section we give implementation details of the network and an evaluation of the proposed TGA in two datasets.

5.1 Implementation Details

In the experiment, we embedded different attention module in the Faster RCNN [24] detector with trident-block.

ResNet50 [7] in bottleneck style is used as the backbone of Faster RCNN detector. For the trident-block, it is constructed as multiple parallel residual blocks by 3×3 convolution with different dilation rates. The multi-branch trident-block placed in the last stage of the Faster RCNN backbone can achieve significant improvement on object detection task, and more than three branches bring no further performance improvement [16]. In our application, although the difference in the scale of the detected object exists but is not particularly huge, we use the multi-branch block with 2 dilated convolutions with the dilation rate of 2 and 3 that share weights, and valid range for each state are set as $[0,220]$, and $[180,\infty]$, to achieve the same representation power on different scale object by different adaptive receptive fields.

We used one state-of-the-art attention module, i.e., SGE [15], as the baseline attention module, and compare it with our proposed TGA. We also tested alone the inter-group attention of TGA, which is proposed in this work. All the involved attention module are based on group attention mechanism. The feature map will be divided into g sub-groups. The value of g affects the semantic distribution between sub-groups and the semantic representation capabilities within the sub-group. If g's value is too large, then the dimension of each group will be reduced, resulting in weak semantic representation ability. On the contrary, if g's value is too small, it is not conducive to the capture of detailed features for each group. In the work of [15], they recommend the number of groups g to be 32 or 64 to boost the performance. In our experiment, we set the number of G to 64 for all attention module. All the attention module is placed after the last BatchNorm layer inside each residual bottleneck on conv1 and conv2 stages.

We used MXNet [1] as our Framework. The image with the original size of 640×480 is used to train the network and the batchsize was set to 1. The backbone of Faster RCNN is pre-trained on Imagenet. We trained the entire network with Stochastic Gradient Descent (SGD) with 0.9 momentum and a weight decay of 0.0001. The learning rate is initialized as 10^{-3} and reduced to 0.5 times every 3 epochs. The network is trained on the GTX 1080Ti GPU for 7 epochs.

5.2 Qualitative Evaluation

We present experimental results for assembling object state detection on the dataset: IKEA-Table and Fender-Assembly. We randomly selected 80% for training and 20% for validation from synthetic datasets, and the testing datasets are all collected from the real-world by a camera.

We demonstrate the performance of the object detector embedded with SGE module, TGA module, and alone the inter-group module of TGA. SGE is not proposed in this work, and act as the baseline attention module in the comparison. The threshold of IOU is selected as 0.5, and the result is summarized in Table 2. Here we don't present the result of the CNNs in the synthetic validation data, because they are already too close to 1. We also visualized the entire tested images with the bounding box and detected state, which can be found in supplementary materials.

First of all, we can notice that all the attention modules improve the performance of CNN. Besides, the CNN with our proposed inter-group attention is also slightly better than with the SGE attention. We think the reason is the inter-group attention focuses more on the spatial relationship between groups, and this spatial relationship is more critical in the object state detection problem. Together with an enhanced representation in the group, the TGA performs best in most case. In the Fender-assembly dataset, the CNN is accurate enough. The failure cases are mostly too tricky due to the strong background illumination. So the TGA and the inter-group attention result with a similar recall rate.

6 Conclusion

In this paper, we deal with the problem of object state estimation, which is critical for AR applications. To address this problem, we present a two level-group attention module (TGA), which consist of inter-group attention and intro-group attention. With our proposed attention module, the relationship cross-feature groups, as well as the representation within a feature group, can be simultaneously enhanced. The TGA can be easily embedded into the CNN layers with only a few extra parameters. The experimental results show the TGA module can enhance the performance on object state estimation, our it outperforms the baseline attention module.

Acknowledgments

This work was partially funded by the INNOPROM Rheinland Pfalz/EFFRE funding program (P1-SZ2-7, 84002637) in cooperation with John Deere GmbH & Co. KG.

References

[1] https://mxnet.apache.org/.

[2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[3] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European conference on computer vision*, pp. 354–370. Springer, 2016.

[4] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

[5] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Dataset		TridentNet [16]			TridentNet [16]+SGE [15]			TridentNet [16]+inter-group			TridentNet [16]+TGA(ours)		
		Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
IKEA-table	State 0	0.783	0.926	0.781	0.882	0.953	0.882	0.845	0.571	0.844	0.870	0.921	0.869
	State 1	0.750	0.785	0.750	0.728	0.845	0.728	0.728	0.840	0.728	0.778	0.828	0.778
	State 2	0.936	0.766	0.936	0.9	0.783	0.9	0.864	0.818	0.864	0.907	0.734	0.907
	State 3	0.558	0.82	0.558	0.728	0.781	0.728	0.673	0.805	0.673	0.735	0.878	0.735
	State 4	0.8115	0.627	0.811	0.770	0.681	0.770	0.877	0.629	0.952	0.770	0.712	0.770
	State 5	0.952	0.571	0.952	0.905	0.613	0.905	0.952	0.625	0.952	0.905	0.826	0.905
	mean	**0.798**	**0.749**	**0.798**	**0.819**	**0.776**	**0.819**	**0.823**	**0.778**	**0.823**	**0.827**†	**0.817***	**0.827**‡
Fender-assembly	State 0	1.0	0.991	0.991	0.953	0.990	0.934	0.972	0.963	0.972	0.962	0.990	0.962
	State 1	0.957	0.957	0.957	0.931	0.991	0.931	0.948	0.982	0.948	0.966	0.991	0.966
	State 2	0.911	0.989	0.910	0.911	0.938	0.910	0.960	0.960	0.960	0.950	0.950	0.950
	State 3	0.898	0.914	0.897	0.880	0.950	0.879	0.926	0.961	0.925	0.926	0.952	0.925
	State 4	0.934	0.970	0.934	0.972	0.928	0.972	0.991	0.929	0.991	0.981	0.945	0.981
	State 5	0.964	0.922	0.964	0.945	0.929	0.945	0.918	0.962	0.918	0.927	0.953	0.927
	mean	**0.944**	**0.957**	**0.942**	**0.932**	**0.956**	**0.928**	**0.953**†	**0.960**	**0.952**‡	**0.952**	**0.964***	**0.952**‡

Table 2: Comparisons of state detection results on the IKEA-table and Fender-assembly. $\dagger, *, \ddagger$ denotes the best mean accuracy, best mean precision and best mean recall, respectively.

[8] T. Hodan, D. Barath, and J. Matas. Epos: Estimating 6d pose of objects with symmetries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11703–11712, 2020.

[9] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

[10] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[11] S. W. Keele and W. T. Neill. Mechanisms of attention. In *Perceptual Processing*, pp. 3–47. Elsevier, 1978.

[12] T. Kong, A. Yao, Y. Chen, and F. Sun. Hypernet: Towards accurate region proposal generation and joint object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 845–853, 2016.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[14] N. Lee, T. Ajanthan, and P. H. Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018.

[15] X. Li, X. Hu, and J. Yang. Spatial group-wise enhance: Improving semantic feature learning in convolutional networks. *arXiv preprint arXiv:1905.09646*, 2019.

[16] Y. Li, Y. Chen, N. Wang, and Z. Zhang. Scale-aware trident networks for object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 6054–6063, 2019.

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pp. 21–37. Springer, 2016.

[18] Y.-F. Ma and H.-J. Zhang. Contrast-based image attention analysis by using fuzzy growing. In *Proceedings of the eleventh ACM international conference on Multimedia*, pp. 374–381, 2003.

[19] N. Minaskan, J. Rambach, A. Pagani, and D. Stricker. Augmented reality in physics education: Motion understanding using an augmented airtable. In *International Conference on Virtual Reality and Augmented Reality*, pp. 116–125. Springer, 2019.

[20] K. Park, T. Patten, and M. Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 7668–7677, 2019.

[21] J. Rambach, C. Deng, A. Pagani, and D. Stricker. Learning 6dof object poses from synthetic single channel images. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 164–169. IEEE, 2018.

[22] J. Rambach, A. Pagani, and D. Stricker. [poster] augmented things: Enhancing ar applications leveraging the internet of things and universal 3d object tracking. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pp. 103–108. IEEE, 2017.

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[24] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.

[25] S. Sabour, N. Frosst, and G. E. Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pp. 3856–3866, 2017.

[26] Y. Su, J. Rambach, N. Minaskan, P. Lesur, A. Pagani, and D. Stricker. Deep multi-state object pose estimation for augmented reality assembly. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 222–227. IEEE, 2019.

[27] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 9627–9636, 2019.

[28] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30. IEEE, 2017.

[29] V. Vasiliu and G. Sörös. Coherent rendering of virtual smile previews with fast neural style transfer. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 66–73. IEEE, 2019.

[30] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

[31] X. Yin, X. Fan, W. Zhu, and R. Liu. Synchronous ar assembly assistance and monitoring system based on ego-centric vision. *Assembly Automation*, 2019.

[32] M. Yuan, S. Ong, and A. Nee. Augmented reality for assembly guidance using a virtual interactive tool. *International journal of production research*, 46(7):1745–1767, 2008.

[33] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, Z. Zhang, H. Lin, Y. Sun, T. He, J. Mueller, R. Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020.

[34] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848–6856, 2018.

[35] H. Zheng, J. Fu, Z.-J. Zha, and J. Luo. Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5012–5021, 2019.