electronics

MDPI

Article

Improving Object Detection Quality by Incorporating Global Contexts via Self-Attention

Donghyeon Lee, Joonyoung Kim and Kyomin Jung *

Department of Electrical and Computer Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Korea; donghyeon@snu.ac.kr (D.L.); kimjymcl@snu.ac.kr (J.K.)
* Correspondence: kjung@snu.ac.kr

Abstract: Fully convolutional structures provide feature maps acquiring local contexts of an image by only stacking numerous convolutional layers. These structures are known to be effective in modern state-of-the-art object detectors such as Faster R-CNN and SSD to find objects from local contexts. However, the quality of object detectors can be further improved by incorporating global contexts when some ambiguous objects should be identified by surrounding objects or background. In this paper, we introduce a self-attention module for object detectors to incorporate global contexts. More specifically, our self-attention module allows the feature extractor to compute feature maps with global contexts by the self-attention mechanism. Our self-attention module computes relationships among all elements in the feature maps, and then blends the feature maps considering the computed relationships. Therefore, this module can capture long-range relationships among objects or backgrounds, which is difficult for fully convolutional structures. Furthermore, our proposed module is not limited to any specific object detectors, and it can be applied to any CNN-based model for any computer vision task. In the experimental results on the object detection task, our method shows remarkable gains in average precision (AP) compared to popular models that have fully convolutional structures. In particular, compared to Faster R-CNN with the ResNet-50 backbone, our module applied to the same backbone achieved +4.0 AP gains without the bells and whistles. In image semantic segmentation and panoptic segmentation tasks, our module improved the performance in all metrics used for each task.

Keywords: object detection; global context; self-attention; convolutional neural network

check for updates

Citation: Lee, D.; Kim, J.; Jung, K. Improving Object Detection Quality by Incorporating Global Contexts via Self-Attention. *Electronics* **2021**, *10*, 90. https://doi.org/10.3390/electronics10010090

Received: 23 November 2020
Accepted: 29 December 2020
Published: 5 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Deep convolutional neural networks (CNN) are now the core of the recognition systems in most computer vision tasks, including classification [1–4], object detection [5–9], semantic segmentation [10,11], and panoptic segmentation [12]. In general, many of the recognition systems adopt a backbone network internally in their framework to extract useful features for their own target tasks. ResNet [3] and Inception [4] are the most popular choices for the backbone networks.

The main goal of object detection is to find as many of objects' tight bounding box locations (regression) and classes (classification) as possible. These two sub-tasks can be well trained by well-known backbone networks, since these networks are originally designed and trained for the classification tasks. However, before the above two sub-tasks (regression and classification), object detectors have to draw some regions of interest (RoI), which is a nontrivial task.

In order to draw some RoIs from an image, two-stage detectors, such as Faster R-CNN [7], propose region proposal networks (RPN) to produce bounding box proposals. In detail, this RPN module receives the feature maps computed by the backbone networks, which are usually designed to have fully convolutional structures. Inefficiency comes from using these feature maps, which are generally highly biased to have local contexts. In this

case, each spatial element (pixel) of the feature maps is inevitably influenced by its local region only. As a result, the RPN module has to draw some RoIs with only local contexts.

The quality of object detectors can be further improved by incorporating global contexts, since some ambiguous objects can be identified by surrounding objects and background. In human perception, an object is recognized by its surroundings and background and how well it matches the global context. For example, in Figure 1, a bright yellow, round object can be the sun when it is in the sky, or a light bulb when it is on a street light, or a sunny-side-up egg when it is on a plate, or the moon when the sky is dark. Therefore, we note that the feature maps should incorporate the global context to detect objects clearly.

Figure 1. Examples of similar objects; the shapes, colors, and textures are similar. It could be difficult for typical feature extractors to clearly capture the characteristics of objects with with local context. In these cases, global context from the entire image could help the detectors to identify objects more clearly.

In this paper, we suggest an efficient way of incorporating global context for the object detection. More specifically, we introduce a self-attention module to incorporate global context to the feature maps computed by the backbone network. The attention mechanism [13] is a well-known technique in natural language processing (NLP) tasks to refine a feature vector (a word) according to other context (a sentence). By applying this self-attention mechanism to the object detection task, the computed feature maps can be refined according to the global contexts. In detail, our proposed self-attention module computes relationships among all elements in the feature maps, and then blends the feature maps considering the computed relationships. With our self-attention module, the RPN can benefit from these refined feature maps with clearer representations using both local and global context.

Researchers have used other methods to incorporate global context into the feature maps. Among them, a global average pooling on the spatial dimensions (width and height) is the most common way to incorporate global context. This is commonly used in image classification tasks [1–4] where computed feature maps are finally global average pooled and then transformed to logits to be classified. In other work [14] for image semantic segmentation tasks, the pyramid scene parsing network (PSPNet) concatenated multiple levels of pyramid features that were average pooled from computed feature maps and then upsampled to original spatial dimensions. In another study [15] for object detection tasks, feature pyramid networks (FPN) were proposed using multiple levels of feature maps, and they propagate some region context through top-down pathway by upsampling. These pyramid methods (PSPNet and FPN) have largely boosted the performance on their

main target tasks, since they can benefit from incorporating some regional context by down-sampling and up-sampling feature maps. However, in this case, global context is indirectly obtained by upsampling, with which it is hard to represent relationships among numerous objects and background elements, whereas our proposed module can directly capture those relationships by the self-attention mechanism.

Note that our proposed self-attention module can be adapted to various CNN backbone networks, such as Inception-ResNet [16], ResNeXt [17], and FBNet [18], so the improvement could be orthogonal to the choice of backbone networks and detection meta-architectures, such as Faster R-CNN [7] and SSD [9]. Furthermore, any computer vision task that can exploit global context may benefit from simply adding our module.

In numerous experiments on the object detection task, our method shows remarkable gains in comparison to other popular models, which use fully convolutional backbone networks. When evaluating Faster R-CNN with the ResNet-50 backbone network on the COCO val2017 (minival) dataset, our proposed self-attention module applied on the same backbone network achieved +4.0 average precision (AP) gains without bells and whistles. When using FPN [15] on the backbone network, which inherently gives some regional context by upsampling, our gain is reduced to +1.4 AP, but it still outperforms the popular models. We also evaluated the self-attention module not only on the object detection task, but also on the semantic segmentation and panoptic segmentation tasks. The experimental results showed that our module improves the performance in all metrics used for each task.

In summary, the main contributions of this paper are listed as follows: (1) We show that incorporating global context can improve the object detection quality further. (2) We propose a self-attention module to blend the feature maps to incorporate global context. (3) We show that the self-attention module can be applied to other computer vision tasks, such as semantic segmentation and panoptic segmentation tasks, to achieve better performance. (4) We suggest some initialization tricks and useful optimizer settings to train the models stably.

2. Related Work

2.1. Object Detectors

With the development of modern deep ConvNets, object detectors such as OverFeat [19] and R-CNN [5] showed dramatic improvements in accuracy. OverFeat adopted a strategy similar to early neural network face detectors by applying a ConvNet as a sliding window detector on an image pyramid. Meanwhile, R-CNN adopted a region proposal-based strategy in which each proposal was scale-normalized before classifying with a ConvNet. Additionally, SPPnet [20] demonstrated that such region-based detectors could be applied much more efficiently on feature maps extracted on a single image scale.

Recent and more accurate detection methods, such as Fast R-CNN [6] and Faster R-CNN [7], advocate using features computed from a single scale, because it offers a good trade-off between accuracy and speed. Those works propose a trainable regional proposal network (RPN) and show great performances on several object detection benchmarks. With a similar philosophy, Mask-R-CNN [10] developed the RPN-based approach for segmentation tasks with ROI align techniques.

RPN-based approaches such as R-CNN series need two stages—one for generating region proposals, one for detecting the object of each proposal, the single shot detector (SSD) [9], and YOLO [8] to take one single shot to detect multiple objects within the image without region proposals generator. On the other hand, two-stage object detectors aim to boost accuracy; YOLO and SSD are tuned for speed but their accuracy trails that of two-stage methods. SSD has a 10–20 lower AP, while YOLO focuses on an even more extreme speed/accuracy trade-off. To achieve better performance in one-stage object detectors, RetinaNet [21] adopts feature pyramid network (FPN) on the backbone network and introduces focal loss.

2.2. Global Contexts and Attention Mechanisms

There have been several works for exploiting global contexts in object detection. RFCN [22] combined a segment proposal and an object detection module for "fully convolutional instance segmentation" (FCIS). The common idea is to predict a set of position-sensitive output channels fully convolutionally [11]. The feature pyramid network (FPN) [15] augments a standard convolutional network with a top-down pathway and lateral connections so the network efficiently constructs a rich, multi-scale feature pyramid from a single resolution input image. Each level of the pyramid can be used for detecting objects at a different scale. Recently, the squeeze-and-excitation network (SENet) [23] introduced an architectural unit that boosts performance at slight computational cost. The main goal is to improve the representational power of a network by explicitly modeling the interdependencies between the channels of its convolutional features.

Attention mechanisms have become an integral part of models that must capture global dependencies [24,25]. In particular, self-attention [26,27], also called intra-attention, calculates the response at a position in a sequence by attending to all positions within the same sequence. Outstandingly, transformer [13] observes that machine translation models could achieve state-of-the-art results by solely using an attention model without any recurrent cells. Recently, variants of transformers have appeared in [28–30], and pre-training with transformers [31] solves numerous NLP tasks, while showing state-of-the-art performance.

3. Network Design

In this section, we introduce the self-attention module to incorporate global contexts that can capture relationships among numerous objects or backgrounds. Here we explain how this module is applied to convolutional neural networks (CNN). In addition, we suggest that the self-attention module have a bottleneck structure that is similar to that used in ResNet [3] to reduce parameters and facilitate stable optimization.

3.1. Theoretical Background for Attention Mechanism

In Natural Language Processing (NLP) tasks, the attention mechanism is a well-known technique to compute a relationship among words in a sentence. Many NLP applications, such as neural machine translation (NMT), extensively use this technique to compute a number of relationships from other words to a word of interest. There are numerous approaches to implement the attention mechanism; the most recent and popular one is defined in the transformer [13] for NMT tasks.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

Generally, as defined in (1), an attention module computes relationships among "query" words (Q) in a target sentence and "key" words (K) of a source sentence. The relationships are computed by an inner product of "query" and "key" words normalized by a square root of dimensions of key d_k with a following softmax operation. Then the attention module blends the "value" of all words (V) in the source sentence by a weighted sum where the weights are the relationships computed by the above softmax operation.

A self-attention mechanism is a special case of the attention mechanism when the source sentence and the target sentence are the same. In this case, an output sentence of a self-attention module is influenced by its own input sentence. In other words, the self-attention module refines the representation of all words in accordance with the representation of its input sentence. This behavior makes sense in numerous NLP tasks, since the meaning of words should be understood by their context. Similarly, in visual tasks like object detection, an object can be determined more clearly by its surrounding contexts and backgrounds. Therefore, based on these ideas, we propose a self-attention module for CNNs that can be applied to most visual tasks, including object detection.

3.2. Self-Attention Modules for Convolutional Neural Networks

Although there are no specific words in images, the self-attention mechanism can be extended to computer vision tasks. In the CNNs, we can regard a spatial element (pixel) on the feature maps as a "visual word" if each spatial element contains sufficient information to explain a certain region in the image.

In popular CNN backbones like ResNets [3], the feature maps become smaller as their stages increase. They usually have five stages that are separated by the resolution of their output feature map. A group of all layers in each stage is called "Cn block" where $n = 1, 2, 3, 4, 5$ and the size of its output feature map is reduced by $2^n \times 2^n$ compared to the size of the input image. Therefore, a spatial element on the output feature map of Cn block corresponds to $2^n \times 2^n$ pixels in the input image.

In the ResNet-C4 block, each spatial element of the output feature map corresponds to 16×16 pixels in the input image. In addition, wider visual receptive fields are obtained by stacks of numerous former convolutional layers. Therefore, each spatial element of the output feature map of ResNet-C4 block can be regarded as a "visual word", since it contains sufficient contexts from large visual receptive fields.

We propose a self-attention module for CNNs to apply the attention mechanism to these visual words. In the self-attention module depicted in Figure 2, each spatial element of the input feature maps is regarded as a visual word and the input feature maps are regarded as visual sentences. The module first transforms each spatial element through query (Q), key (K), and value (V) dense layers. Then it computes relationships among all visual words by matrix multiplication with Q, and K. These relationships are exploited as weights when the self-attention module blends the input feature maps by the weighted sum of itself. Through these operations, every spatial element of feature maps is influenced by all other spatial elements. In other words, local contexts are refined considering global contexts by directly computing all of the relationships among local contexts.

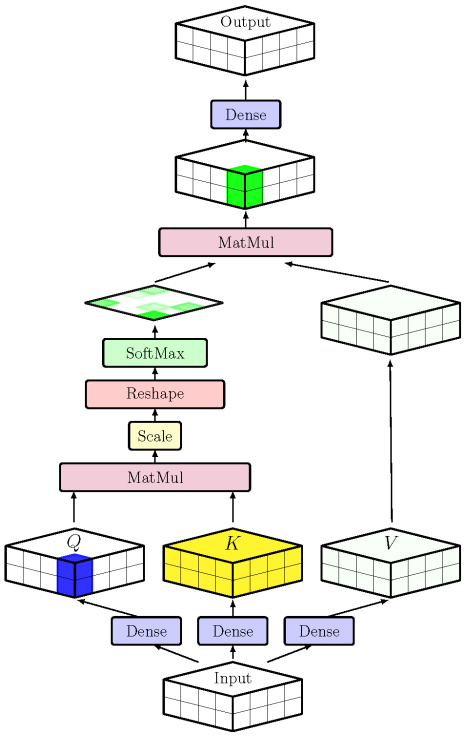

Figure 2. A self-attention module to incorporate global context. An input feature map is transformed to query Q, key K, and value V by fully connected layers. This self-attention module computes relationships among all elements of feature maps by query Q and key K, and then value V is blended by itself considering the computed relationships. Lastly, the fully connected layer is applied to recover channel dimensions and give the module a bottleneck structure.

There is a slight difference in the attention mechanism between NLP tasks and visual tasks. The attention mechanism for NLP tasks is originally defined on the 3D features shaped by (N, T, C) where N, T, and C denote the batch, length, and depth dimensions, respectively. On the other hand, the feature maps of CNNs are composed of 4D features, (N, H, W, C) where N, H, W, and C denote the batch, height, width, and depth dimensions, respectively. In the 4D feature maps, the visual words are considered to be distributed on the spatial dimensions. Due to the different dimensions of feature representations, the self-attention module for CNNs needs to reshape its input and output feature maps. There is no harm in combining spatial dimensions H and W, since the contexts of visual words can be safely kept in their combined dimensions. Therefore, in the implementation of the self-attention module for CNNs, 4D visual feature maps (N, H, W, C) are transformed to 3D feature maps (N, HW, C), as in sentence representation, and then the self-attention mechanism can be applied. Finally, outputs of the self-attention module are reverted back to 4D feature maps (N, H, W, C) like usual visual feature maps. Note that this implementation does not affect the actual computation of the attention mechanism, as it just changes the order of visual words.

3.3. Backbones with Self-Attention Modules

The standard method of using the backbone network for the Faster R-CNN [7] architecture is to divide the backbone network to two sets of blocks {C1, . . . , C4}, and {C5}. Faster R-CNN jointly trains the region proposal network (RPN) with the former blocks and trains the Fast R-CNN [6] with the latter block. The RPN takes the output feature map of the C4 block as inputs to generate bounding box proposals, and the Fast R-CNN [6] takes the pooled features of corresponding bounding box proposals from the output feature map of the C4 block as inputs to classify the bounding box proposals. The overall structures are depicted in Figure 3. Note that the RPN uses the output feature map of the C4 block as input, and the Fast R-CNN uses the same feature map as input by pooling through "RoIPool" (or "RoIAlign") operators.

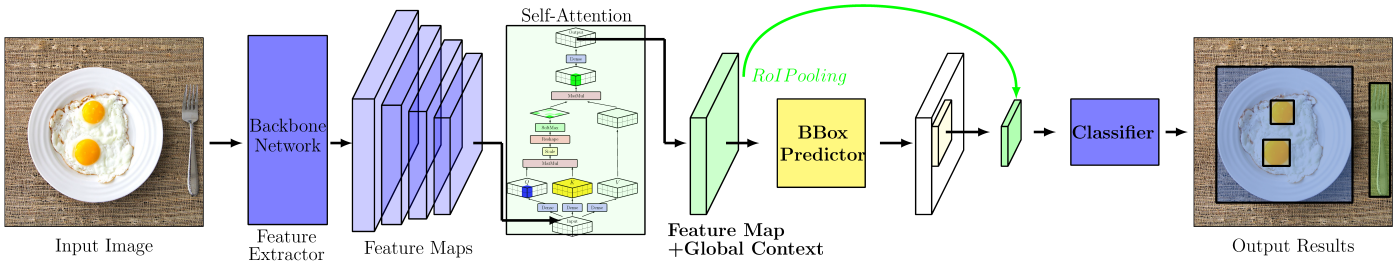

Figure 3. An overall structure for the object detection pipeline with the self-attention module. The self-attention module is applied to the top of the feature map computed by backbone networks. In the Faster R-CNN [7]; this enables the RPN to exploit both local and global contexts when generating bounding box proposals and forwarding features by "RoIPool" (RoIAlign) operations.

We claim that the most appropriate place to apply the self-attention module to the backbone network is after the C4 block, before the RPN generates the bounding box proposals. When the RPN is detecting objects from the entire image, considering both local and global contexts can be more helpful than considering only local contexts. Therefore, we locate the self-attention module between the C4 block and the RPN to incorporate the global contexts before the RPN generates the bounding box proposals. In this case, the RPN can generate better bounding box proposals by exploiting local and global contexts, which leads to better performance on the object detection task. This network design does not harm the extraction of local contexts at lower stages of the backbone, and it helps to generate global contexts with high-level features.

In the case of using feature pyramid networks (FPN) [15] on the backbone network, we apply the self-attention module on the top of the backbone network depicted in Figure 4. The FPN exploits all stages of the backbone network as inputs {C2, . . . , C5} that are used to

generate a feature pyramid {P2, . . . , P5}. In the FPN, there is a top-down pathway through {P5, . . . , P2} that allows it to propagate the higher-level rich features to lower stages. We apply the self-attention module on the top of the backbone network, which locates the self-attention module between the C5 and P5 feature maps. In this case, P5 feature maps acquire global contexts by the self-attention module, and then global contexts are propagated through the top-down pathway. We can add more self-attention modules among lower {C2, . . . , C4} and {P2, . . . , P4} feature maps, but the best performance is achieved when the self-attention module is located between C5 and P5 in our experiments.

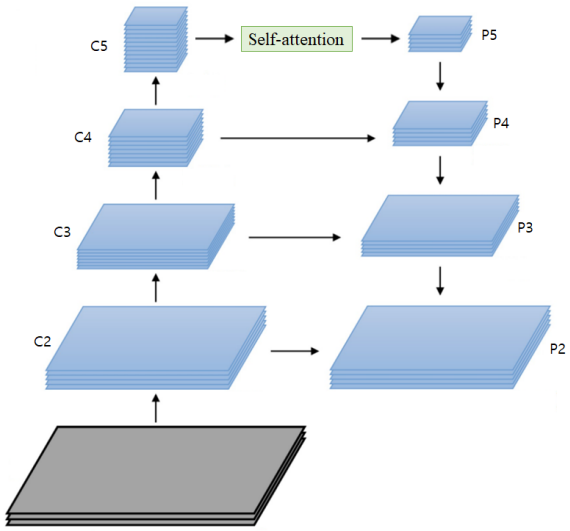

Figure 4. Feature pyramid networks (FPN) [15] with the self-attention module. The self-attention module is applied on the top of the backbone network. Global contexts are acquired by the self-attention module and propagated through the top-down pathway of the FPN. All FPN stages are influenced by the global contexts, which helps the RPN to generate better bounding boxes.

3.4. Bottleneck Structure

A standard attention module contains three fully connected layers that compute query (Q), key (K), and value (V) from input feature maps. These fully connected layers can change their channel dimension (depth) when needed. In our proposed self-attention module, we introduce one more output fully connected layer to give the module the bottleneck structure widely used in ResNets. Our bottleneck module operates as the same as that of ResNets when expanding and contracting the channel dimension. In our implementation of the bottleneck structure, the number of channels of query, key, and value feature maps is reduced by four times in the channel dimension, and then the number of channels of the self-attention outputs is reverted to its original number.

This bottleneck structure has several advantages when computing feature maps. One of the advantages is that the amount of computations and parameters of self-attention module can be reduced. In addition, the most important advantage we found is that it facilitates stable optimization. Empirically, we have struggled with the optimization of self-attention modules, because the loss often diverges when the parameters of the self-attention module are randomly initialized. We have tried several well-known initialization methods, but most of them do not change this situation. However, with the bottleneck structure, we can use pre-trained weights of ResNets to initialize the self-attention module, since ResNets have the same bottleneck structure. We have observed that it is successfully finetuned from the pre-trained weights of ResNet blocks for initializing self-attention modules with common optimization settings of Faster and Mask R-CNN [7,10] training.

3.5. Translation Equivariance Characteristic

We employ the self-attention module at the last stage of feature maps of the backbone network, right before the RPN. As stated in Section 3.3, one reason for the location is that

the self-attention module should be applied to visual words that have large visual receptive fields. In addition, another important reason is that the all object detectors must follow the translation equivariance characteristic. That is, the translation of inputs should be forwarded to the equal amount of the translation on the outputs. In other words, if the location of an object is moved (translation), then the location of the detection should be moved by an equal amount (equivariance).

To obtain the translation equivariance characteristic, most of the conventional object detectors are designed to have fully convolutional structures since the translation equivariance characteristic can be easily achieved by local contexts. However, in our network design, the self-attention mechanism obtains global contexts at the cost of losing a portion of local contexts. Due to this, stacking the self-attention modules multiple times can lead to the loss of local contexts, decreasing the performance. Rather, we employ the self-attention module once before the RPN to exploit both local and global contexts when detecting objects.

4. Experiments

4.1. Object Detection

We performed a variety of experiments on on 2017 COCO detection datasets [32] with 118,000 training images and 5000 validation images. We used meta-architectures of object detectors as Faster R-CNN [7] for the two-stage detection architecture and RetinaNet [21] for the one-stage detection architecture. For the backbone networks, we used ResNet-50 and ResNet-101 by finetuning the pre-trained weights from the ImageNet classification checkpoint. Since the model was trained with small batch sizes, each batch norm layer on the backbone networks was frozen during training. All experiments were done using TensorFlow Object Detection API [33]. Note that the height and width of input images were resized to have minimum 600 pixels and maximum 1024 pixels, which are the default settings in the API. Most of the other settings were the same as standard use which are defined in the API. Some recent papers [15,21,34,35] report their performances with higher numbers in AP (e.g., about 35 AP for Faster R-CNN with ResNet-50 backbone), but direct comparisons are unfair since their input images are much bigger (about $\times 1.6$) and they use multi-scale training. Those large performance gaps come from the image size, since small objects are clearer in large images. We followed the default settings in the API, where the benchmarks are publicly available in the model zoo page of the TensorFlow Object Detection API website.

Metric

For the object detection, AP (average precision averaged over categories and IoU thresholds) [32] is the primary metric and $AP_{@.50}$ and $AP_{@.75}$ are additional metrics (at a single IoU threshold of 0.5 and 0.75 respectively). We also report COCO AP on objects of small, medium, and large sizes (namely, AP_s, AP_m, AP_l). Next we evaluate the average recall (namely, AR and AR) on small, medium, and large objects (namely, AR_s, AR_m, and AR_l). We report results for 100 proposals per images (AR^{100}) [36].

4.2. Global Contexts

To verify the effectiveness of incorporating global contexts by the self-attention module, we trained the Faster R-CNN [7] and RetinaNet [21] with ResNet-50 and ResNet-101 backbone networks. Experimental results are shown in Table 1 and Table 2 respectively. We observe that incorporating global contexts improve the object detection quality much further. Especially in the experiment of Faster R-CNN with ResNet-50 backbone network, the performance improves by +4.0 AP after adding the self-attention module.

In the experiments of training RetinaNet [21], which uses SSD [9] as meta-architecture with ResNet and FPN [15] backbone networks, the detection performance improved by 1.4 AP. The performance gain is reduced compared to the backbone without FPN, but our model still outperforms the RetinaNet models. The performance gap can be reduced when FPN is used in backbone networks since FPN can capture a portion of region contexts.

FPN forwards rich high-level features to the lower stage feature maps in the pyramid via upsampling, which allows lower stage feature maps to be influenced by some region contexts. At this point, the self-attention module can fill up the rest of global contexts by directly computing relationships among all elements, leading to improve the object detection quality further.

Table 1. Object detection results using **Faster R-CNN** [7] on the COCO `val2017` dataset. Here ResNet is abbreviated as "R". Incorporating global contexts (GC) improves the quality of object detection in all metrics. The number in bold means superiority compared to that of the baseline in the metric.

Faster R-CNN	AP	AP_s	AP_m	AP_l	$AP_{@.5}$	$AP_{@.75}$	AR^{100}	AR_s^{100}	AR_m^{100}	AR_l^{100}
(a) R-50 [3]	30.1	7.65	27.2	45.7	49.7	31.3	46.6	64.2	44.9	19.0
(b) R-50 + GC	**34.1**	**10.1**	**31.2**	**49.6**	**54.3**	**35.5**	**49.8**	**66.8**	**48.6**	**23.9**
(c) R-101	33.9	8.93	30.7	51.0	53.5	35.6	49.7	66.6	48.0	21.6
(d) R-101 + GC	**35.9**	**10.9**	**32.9**	**52.8**	**56.1**	**37.7**	**51.5**	**68.8**	**50.6**	**23.6**

Table 2. Object detection results using **RetinaNet** [21] on the COCO `val2017` dataset. Here ResNet is abbreviated as "R". Note that RetinaNet uses FPN [15] with a backbone network by default. Incorporating global contexts (GC) improves the quality of object detection in all metrics. The number in bold means superiority compared to that of the baseline in the metric.

RetinaNet	AP	AP_s	AP_m	AP_l	$AP_{@.5}$	$AP_{@.75}$	AR^{100}	AR_s^{100}	AR_m^{100}	AR_l^{100}
(a) R-50-FPN	35.9	11.9	33.5	50.7	53.7	38.8	53.3	25.6	53.5	68.9
(b) R-50-FPN + GC	**37.3**	**14.3**	**35.2**	**51.5**	**55.9**	**40.4**	**54.5**	**28.4**	**54.3**	**69.2**
(c) R-101-FPN	37.5	13.2	35.6	53.1	55.5	40.8	54.3	27.1	54.4	69.9
(d) R-101-FPN + GC	**38.4**	**14.6**	**36.3**	**53.6**	**56.8**	**41.8**	**55.5**	**28.6**	**55.7**	**70.8**

In addition, the large portion of the performance gap comes from detecting small objects. They are more difficult to be captured than large objects since RoIPooled features of small bounding boxes inherently lacks of contexts and their receptive fields are very small. At this point, the self-attention module fills a large amount of global contexts from other spatial regions, and helps small objects to be identified more clearly.

4.3. Self-Attention Modules

We explore several structural choices for the proposed self-attention modules. The self-attention module has a block structure, and it can be stacked in multiple times to intensify the influence of global contexts to the feature maps. In addition, the self-attention module supports multi-head attention to split attention along several sets of channel dimensions rather the entire channel dimensions. Experimental results are shown in Table 3.

We observe that stacking self-attention modules in multiple times can hurt the object detection quality. Comparing the results of (layer $\times$ head): $\{(1 \times 16), (2 \times 16), (3 \times 16)\}$, the performance degenerates when stacking more self-attention modules. As mentioned in Section 3.5, if we enhance the feature representation with the global contexts too much, the performance can be degenerated, since object detectors should have the translation equivariance characteristic. The local context involves a large portion of information in a certain region, so the local contexts should not be encroached on much by the global contexts.

In the experiments of multi-head attention, a small number of multi-heads can be enough to improve the performance. We observe that four or eight heads are proper to learn multi-head attention. In case when stacking multiple self-attention modules, more heads can fill up some lost performance but cannot fully recover the performance of the case when not stacking self-attention module.

Table 3. Results using **Faster R-CNN** [7] on the COCO `val2017` dataset varying a number of self-attention heads.

Setting	Layer × Head	AP
Faster R-CNN	1 × 1	33.1
	1 × 2	33.8
	1 × 4	34.1
	1 × 8	33.7
	1 × 16	33.5
ResNet-50 + GC	2 × 2	27.3
	2 × 4	31.7
	2 × 8	31.5
	2 × 16	32.2
	3 × 16	31.7

4.4. Semantic and Panoptic Segmentation

The self-attention module can be applied to any convolutional neural networks, so other vision tasks can be improved by incorporating global contexts. Other well-known visual tasks are image semantic segmentation task and recently defined panoptic segmentation [12] task. When the self-attention module is applied to the backbone network of two tasks, the performances are improved in all metrics of their tasks. The results are shown in Tables 4 and 5.

Table 4. Image semantic segmentation results on the COCO `val2017` dataset. Here ResNet is abbreviated as "R". Incorporating global contexts (GC) improves the quality of image semantic segmentation in all metrics. The number in bold means superiority compared to that of the baseline in the metric.

Backbone	mIoU	fwIoU	mACC	pACC
(a) R-50-FPN	41.2	68.5	52.2	80.3
(b) R-50-FPN + GC	**42.6**	**69.3**	**53.7**	**80.8**

Table 5. Panoptic segmentation results on the COCO `val2017` dataset. Here ResNet is abbreviated as "R". Incorporating global contexts (GC) improves the quality of panoptic segmentation in all metrics. The number in bold means superiority compared to that of the baseline in the metric.

Backbone	PQ	SQ	RQ	PQ^{St}	PQ^{Th}
(a) R-50-FPN	39.4	77.8	48.3	29.6	45.9
(b) R-50-FPN + GC	**40.2**	**78.1**	**49.3**	**31.0**	**46.4**

Metric

We report standard semantic and instance segmentation metrics for the individual tasks using evaluation code provided by each dataset. For semantic segmentation, the mIoU (mean Intersection over Union) [37] is the primary metric on the COCO dataset [32]. We also report fwIoU (frequency weighted IoU). For panoptic detection, we adopt PQ (Panoptic Quality) [12] as the primary metric. PQ captures both recognition and segmentation quality, and treats both stuff and thing categories in a unified manner. Additionally, we use PQ^{St} and PQ^{Th} to report stuff and thing performance separately.

4.5. Optimization

We observe that it is hard to train the models stably with randomly initialized parameters, which can screw up the rich feature representation from pre-trained backbone networks. In that case, the training loss often easily diverges or converges with inferior results. This behaviour is also reported in other recent researches [34,38] where they try to learn models from randomly initialized parameters with large learning rates.

To solve this, we trained the model with Adam [39] optimizer first and switched to the common stochastic gradient descent (SGD) optimizer, which is introduced in [40]. We empirically found that the Adam optimizer is able to train the randomly initialized weights without divergence. Finally, we finetuned the models by SGD optimizer to boost the performance. The evaluation curves of the combination of Adam and SGD optimizers are shown in Figure 5.

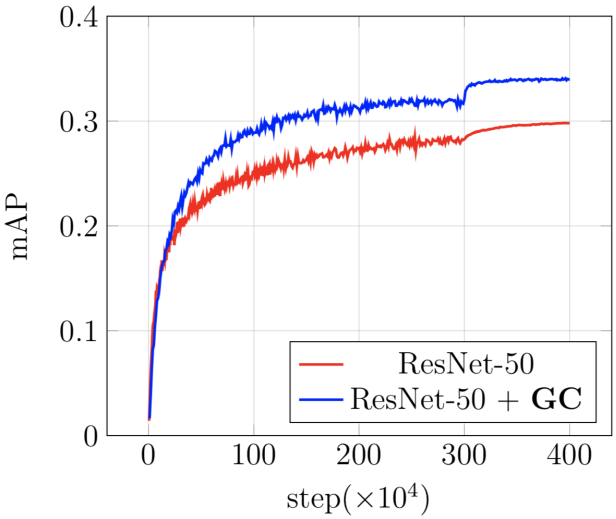

Figure 5. Comparisons of evaluation curves on the COCO `val2017` dataset with Adam + SGD optimizer.

In addition, we adopted a bottleneck design to perform an initialization trick. Rather than training from randomly initialized parameters, we finetune the pre-trained parameters from ResNet classification checkpoint. Since fully-connected layers are considered as a 1×1 convolution, the parameters of fully connected layers in the self-attention module can be initialized from the 1×1 convolutional layer in the bottleneck block in ResNet. The optimization results are shown in Table 6.

Table 6. Comparisons between two optimizers, Adam + SGD and SGD with momentum, on the COCO `val2017` dataset. Here ResNet is abbreviated as "R". The number in bold means superiority compared to that of the baseline in the metric. When training Faster R-CNN with ResNet + GC backbone, Adam optimizer and finetuning with SGD optimizer can successfully train the model while SGD optimizer with momentum diverges (marked as ×). In the case of training RetinaNet, which uses FPN by default, SGD with momentum optimizer can train the model successfully. It seems that FPN helps the optimization stabilize.

Detector	Backbone	Optimizer	AP
Faster R-CNN	R-50	Adam + SGD	30.1
		Momentum	30.0
Faster R-CNN	R-50 + GC	Adam + SGD	**34.1**
		Momentum	×
Faster R-CNN	R-101	Adam + SGD	33.9
		Momentum	32.0
Faster R-CNN	R-101 + GC	Adam + SGD	**35.9**
		Momentum	×
RetinaNet	R-50-FPN	Adam + SGD	35.2
		Momentum	35.9
RetinaNet	R-50-FPN + GC	Adam + SGD	35.4
		Momentum	**37.3**

We observe that longer training steps can boost performance since there is more space to optimize the attention parameters. As studied in [35], when training shortly, training

from scratch can be worse than finetuning from pre-trained weights, however it can catch up with longer training steps. The evaluation curves shown in Figure 6 support that claim and this technique is also applied when training the self-attention module. In addition, the qualitative comparisons for the use of the self-attention module are shown in Figure 7.

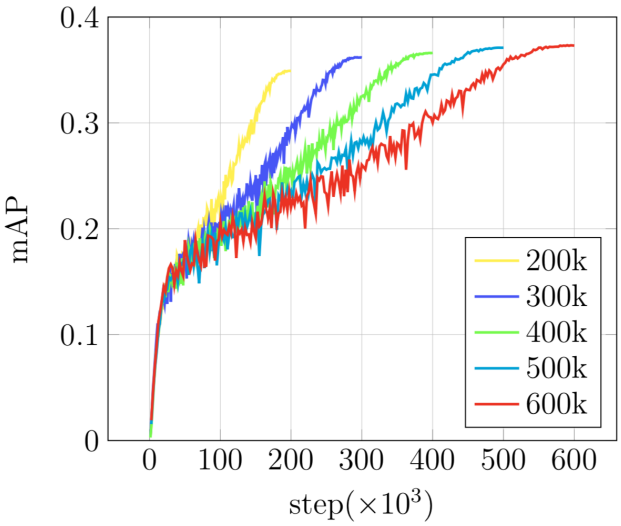

Figure 6. Comparisons of evaluation curves with longer training steps. RetinaNet with ResNet-50 backbone was trained by a momentum optimizer with cosine decay and evaluated on the COCO `val2017` dataset.

Figure 7. *Cont.*

Figure 7. Qualitative results on the COCO `val2017` dataset of Faster R-CNN meta-architecture with ResNet-101 + GC backbone. Detection results are on the left side of image pairs, and corresponding groundtruths are on the right side of the pairs.

5. Conclusions

We explored the influences of global contexts from attention layers in modern object detectors. We propose a self-attention module for CNNs in order to capture relationships among other visual words indicating objects or backgrounds. We suggest a network design for object detectors to apply the self-attention module effectively. From the experimental results, our network design improves the detection performance largely in terms of AP. This results indicate that the attention mechanisms defined in NLP tasks also have a potential to improve performance in computer vision tasks. We verify its potential in the semantic segmentation task and the panoptic segmentation task.

We believe that visual words in computer vision tasks are not that different from real words in NLP tasks. The visual words can act as usual textual words if visual words have sufficient contexts extracted from large visual receptive fields. From the help of pre-trained CNNs, visual words can represent regional contextualized information of where they are located. Therefore, self-attention mechanisms for visual words can improve the representational power of visual features. Any computer vision task that can exploit relationships among objects or backgrounds should benefit by applying attention mechanisms. We hope that this work facilitates the use of attention mechanisms even over computer vision tasks to multi-modal tasks.

Author Contributions: Conceptualization, D.L. and J.K.; methodology, D.L.; software, D.L.; validation, D.L. and J.K.; formal analysis, D.L.; investigation, D.L. and J.K; resources, K.J.; data curation, D.L.; writing—original draft preparation, D.L. and J.K; writing—review and editing, K.J.; visualization, J.K.; supervision, K.J.; project administration, K.J.; funding acquisition, K.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Samsung Electronics. This work was also supported by the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2020. This work was also supported by the Department of Electrical and Computer Engineering, Seoul National University, and Automation and Systems Research Institute (ASRI), Seoul National University.

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.

Abbreviations

The following abbreviations are used in this manuscript:

mAP	mean average precision
RoI	regions of interest
RPN	region proposal network
NLP	natural language processing
FPN	feature pyramid network
SSD	single shot multibox detector
R-#	ResNet-#
GC	global contexts
SGD	stochastic gradient descent

References

1. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.-F. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
2. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
3. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
4. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
5. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
6. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
7. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
9. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*; Springer: Berlin, Germany, 2016; pp. 21–37.
10. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
11. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
12. Kirillov, A.; He, K.; Girshick, R.; Rother, C.; Dollár, P. Panoptic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 9404–9413.
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
14. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
15. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
16. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
17. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
18. Wu, B.; Dai, X.; Zhang, P.; Wang, Y.; Sun, F.; Wu, Y.; Tian, Y.; Vajda, P.; Jia, Y.; Keutzer, K. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 10734–10742.
19. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv* **2013**, arXiv:1312.6229.
20. Purkait, P.; Zhao, C.; Zach, C. SPP-Net: Deep absolute pose regression with synthetic views. *arXiv* **2017**, arXiv:1712.03452.
21. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
22. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 379–387.

23. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
24. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
25. Yang, Z.; He, X.; Gao, J.; Deng, L.; Smola, A. Stacked attention networks for image question answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 21–29.
26. Cheng, J.; Dong, L.; Lapata, M. Long short-term memory-networks for machine reading. *arXiv* **2016**, arXiv:1601.06733.
27. Parikh, A.P.; Täckström, O.; Das, D.; Uszkoreit, J. A decomposable attention model for natural language inference. *arXiv* **2016**, arXiv:1606.01933.
28. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.V.; Salakhutdinov, R. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *arXiv* **2019**, arXiv:cs.LG/1901.02860.
29. Kitaev, N.; Łukasz Kaiser.; Levskaya, A. Reformer: The Efficient Transformer. *arXiv* **2020**, arXiv:cs.LG/2001.04451.
30. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The Long-Document Transformer. *arXiv* **2020**, arXiv:cs.CL/2004.05150.
31. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:cs.CL/1810.04805.
32. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*; Springer: Berlin, Germany, 2014; pp. 740–755.
33. Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S.; et al. Speed/accuracy trade-offs for modern convolutional object detectors. *arXiv* **2017**, arXiv:cs.CV/1611.10012.
34. Peng, C.; Xiao, T.; Li, Z.; Jiang, Y.; Zhang, X.; Jia, K.; Yu, G.; Sun, J. Megdet: A large mini-batch object detector. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6181–6189.
35. He, K.; Girshick, R.; Dollár, P. Rethinking imagenet pre-training. In Proceedings of the IEEE International Conference on Computer Vision, Long Beach, CA, USA, 16–20 June 2019; pp. 4918–4927.
36. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
37. Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
38. Goyal, P.; Dollár, P.; Girshick, R.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv* **2017**, arXiv:1706.02677.
39. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
40. Keskar, N.S.; Socher, R. Improving generalization performance by switching from adam to sgd. *arXiv* **2017**, arXiv:1712.07628.