Dual-stream Network for Visual Recognition

Anonymous Author(s)
Affiliation
Address
email

Abstract

Transformers with remarkable global representation capacities achieve competitive results for visual tasks, but fail to consider high-level local pattern information in input images. In this paper, we present a generic Dual-stream Network (DS-Net) to fully explore the representation capacity of local and global pattern features for image classification. Our DS-Net can simultaneously calculate fine-grained and integrated features and efficiently fuse them. Specifically, we propose an Intra-scale Propagation module to process two different resolutions in each block and an Inter-Scale Alignment module to perform information interaction across features at dual scales. Besides, we also design a Dual-stream FPN (DS-FPN) to further enhance contextual information for downstream dense predictions. Without bells and whistles, the propsed DS-Net outperforms Deit-Small by 2.4% in terms of top-1 accuracy on ImageNet-1k and achieves state-of-the-art performance over other Vision Transformers and ResNets. For object detection and instance segmentation, DS-Net-Small respectively outperforms ResNet-50 by 6.4% and 5.5 % in terms of mAP on MSCOCO 2017, and surpasses the previous state-of-the-art scheme, which significantly demonstrates its potential to be a general backbone in vision tasks.

1 Introduction

In recent years, convolutional neural networks (CNNs) have dominated various vision tasks including but not limited to image recognition [17, 13, 30, 31, 4, 14, 32, 15, 41], object detection [29, 1, 21, 28, 19, 40, 24, 16, 9] and segmentation [12, 23], thanks to their unprecedented representation capacity. However, limited receptive fields of convolutions inevitably neglect the global patterns in images, which might be crucial during inference. For example, it is more likely a chair than a elephant nearby a table. Such object-level information cannot be fully explored by local convolutions, that hampers further improvement of CNNs. Motivated by the success of Transformer architecture [34] in Natural Language Processing (NLP), researchers are trying to apply Transformer to vision tasks and have obtained promising results. In [8, 43, 38, 6, 35, 36, 22, 2, 42], many novel architectures and optimizing strategies are proposed and achieve comparable or even better performance than CNNs, emphasizing the significance of extracting global features in vision tasks.

Inspired by the verified efficacy of CNN and Transformer, many concurrent works, such as ContNet [37], CrossViT [3], CvT [36], and PVT [35], attempt to introduce convolutions to vision transformers in different manners, hoping to combine both advantages. In CvT [36] and PVT [35], linear projection of self-attention block in the Transformer module are replaced with convolutional projection. In contrast, ContNet[37] performs convolution on different token maps to build mutual connections, which is similar to Swin Transformer [22].

However, some drawbacks of existing models still remain to be solved. Firstly, works mentioned above either perform convolution and attention mechanisms sequentially, or just replace linear projection with convolutional projection in attention mechanisms, which obviously cannot be the

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

most ideal design. Additionally, the conflicting properties of such two operations, convolution for local patterns but attention for global patterns, might cause ambiguity during training, which prevents their merits from merging to the maximum extent. Furthermore, self-attention can capture long-range information via the built-in all-pair interaction in theory, but it is quite possible that attentions might be confused and disturbed by neighboring details in high resolution feature maps, fail to build up object-level global patterns(see Fig 1(a)). Finally, the computation cost of self-attention is unaffordable due to the quadratic computation complexity of sequence length. Although PVT [35] and APNB [44] downsample the key-query features to improve the efficiency of self-attention operator, they both abandon fine-graind local details of the image, which greatly impairs their performance.

In this paper, we address the these issues by introducing a *Dual-stream Network (DS-Net)*. Instead of single stream architecture as previous works, our DS-Net adopts Dual-stream Blocks (DS-Blocks), which generates two feature maps with different resolutions, and retains both local and global information of the image via two parallel branches. We propose a Intra-scale Propagation module here to process two feature maps. Specifically, high-resolution features are used to extract fine-grained local patterns with depth-wise convolution, while low-resolution features are expected to summarize long-range global patterns. Considering low-resolution features themselves contain more integrated information, it would be much more easier for self-attention mechanism to capture object-level patterns rather than overwhelmed by trivial details (see Fig 1(b)). Such dual-stream architecture disentangles local and global representations, which helps maximize both their merits, and thus generates better representations compared to Deit baseline (see (c) and (e) in Fig 1). Besides, low-resolution feature maps for self-attention dramatically reduce the memory cost and computation complexity. After parallelly processing dual streams, we present Inter-scale Alignment module based on co-attention mechanism at the end of DS-Blocks. This is because local details and global patterns capture different perspectives of the image, which are misaligned not only in pixel positions but also in semantics. Hence, this module is designed for modeling complex cross-scale relations and adaptively fuse local and global patterns.

Besides, we apply our DS-Blocks to Feature Pyramid Networks for further feature refinement, named DS-FPN. In this way, multi-level features are capable of extracting contextual information from both local and global views, improving performance of downstream tasks. This demonstrates that our Dual Stream design could be utilized as a plug-in building block not only for image recognition, but for many other vision tasks.

The contributions of this work are concluded as follows:

1. We present a novel Dual-Stream Network, named DS-Net, which retains both local and global features in DS-Block. The independent propagation maximize the advantages of convolution and self-attention, thus eliminating the conflicts during training.
2. We propose Intra-scale Propagation and Inter-Scale Alignment mechanism to achieve effective information flows within and between features of different resolutions, thus generating better representations of the image.
3. We introduce Dual-stream Feature Pyramid Network (DS-FPN) to enhance contextual information for downstream dense tasks and achieves better performance with little extra costs.
4. Without bells and whistles, the proposed DS-Net outperforms Deit-small baseline by significant margins in terms of top-1 accuracy on ImageNet-1k and achieves state-of-the-art performance over other Vision Transformers and CNN-based networks on image classification and downstream tasks, including object detection and instance segmentation.

2 Related work

Vision Transformers. Motivated by the great success of Transformer in natural language processing, researchers are trying to apply Transformer architecture to Computer Vision tasks. Unlike mainstream CNN-based models, Transformer is capable of capturing long-distance visual relations by its self-attention module and provides the paradigm without image-specific inductive bias. ViT [8] views 16 $\times$ 16 image patches as token sequence and predicts classification via a unique class token, which shows promising results. Subsequently, many works, such as DeiT [33] and PVT [35] achieve further improvement on ViT, making it more efficient and applicable in downstream tasks. Besides,

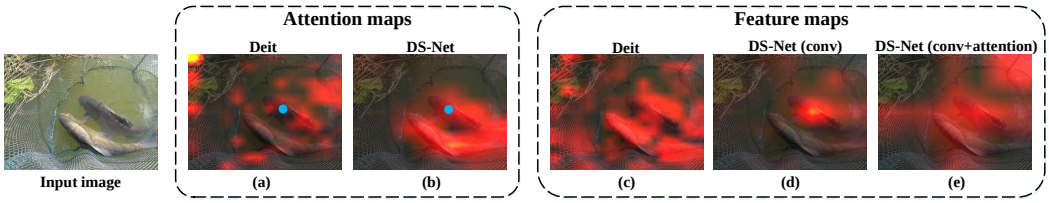


Figure 1: Comparison of attention maps and feature maps of Deit and our proposed DS-Net, both of which are obtained from the last block. (a) and (b) are attention weights of the blue points in the images. To achieve clearer illustration, the figures are acquired by overlaying heatmaps to the input images.

models based on Transformer also give leading performance in many other vision tasks such as object tracking and video analysis.

Local and Global Features in Transformers. Despite the Transformer's superiority upon extracting global representations, image-level self-attention is unable to capture fine-grained details. To tackle this issue, previous works assert to obtain the local information at the same time and then fuse features of the two scales. TNT [11] proposes intra-patch self-attention block to model local structures and aggregates them with global patch embeddings. Swin Transformer [22] extracts local features within each partitioned window and fuses them by self-attention in successive shifted windows. Similarly, shifted windows are replaced by convolutions in ContNet [37]. Following such design of local windows, Twins [5] uses inter-window attention to aggregate global features. Swin Transformer and Twins both give advanced performance, but still rely on local-window self-attention to capture fine-level features, which works intuitively worse than CNN. Besides, Swin's shifted windows are complex for device optimization and Twins [5] recurrently local and global attentions would constrain each other's representation learning.

Equip Transformers with Convolution. Compared to self-attention mechanism, CNN has its unique advantages of local modeling and translation invariance. Therefore, some existing transformers explore the hybrid architecture to incorporate both merits for better visual representation. T2T [38] progressively aggregate neighboring tokens before transformers via overlapping convolution-alike way to enhance their representations. Followingly, CvT [36] designs Convolutional Token Embedding and Convolutional Transformer Block for capturing more precise local spatial context. CPVT [6] utilizes one convolutional layer to dynamically generate positional encodings, adaptive for varying image scales. Conformer [27] combines transformer with an independent CNN network for specializing local features and flows information via lateral connections. Nonetheless, none of the aforementioned models maximizes the potential of the hybrid architecture. They either use CNN for marginal operations to boost Transformer, *e.g.*, tokens generation in CvT [36] and positional encodings in CPVT [6], or view CNN as a relatively separate branch, like Conformer.

Different from previous works, in proposed DS-Blocks, we treat self-attention and convolution as dual resolution processing paths via Intra-scale Propagation module, where the former aims to extract local fine-grained details, and the latter focus on exploring features from a global view. On top of that, their information is reasonably fused by Inter-scale Alignment module based on co-attention, which addresses the issue of feature misalignment. By this design, the local and global patterns could be simultaneously extracted and the capability of CNN is fully exerted in Transformer architecture.

3 Method

3.1 Dual-scale Representations

The overall pipeline of our proposed DS-Net is shown in Fig 2. Motivated by stage-wise design in previous networks like ResNet [13], we set 4 stages in the architecture, whose down-sampling factors are 4, 8, 16, 32, respectively. Within each stage, Dual-stream Blocks (DS-Blocks) are adopted to generate and combine dual-scale representations.

Our key idea is to keep local features in a relatively high resolution to reserve local details, while represent global features with a lower resolution (1/32 of the image size) to retain global patterns.

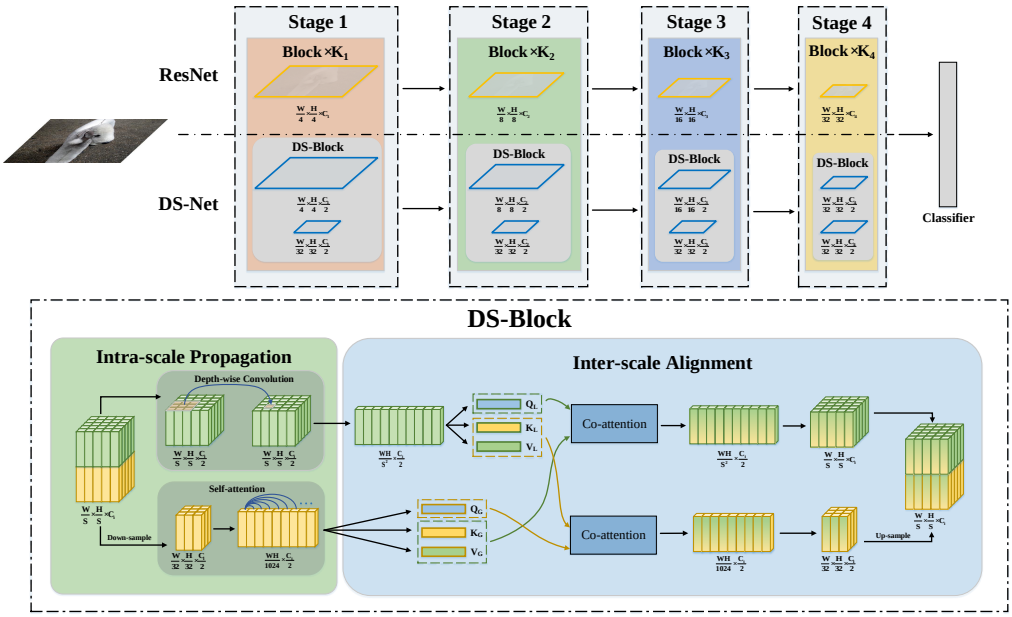


Figure 2: Illustration of the proposed DS-Net, including Intra-scale Propagation module and Inter-scale Alignment module. Compared to ResNet, in which only single resolution is processed, our DS-Net, instead, generates dual-stream representations via DS-Blocks.

Specifically, in each DS-block, we split the input feature map into two parts at the channel dimension. One is for extracting local features, denoted as f_l, the other is for summarizing global features, denoted as f_g. Notably, we keep the size of f_g unchanged in all stages across the network by down-sampling it with a proper factor, $\frac{W}{32} \times \frac{H}{32} \times \frac{C_i}{2}$, where W and H represents the width and height of the input image, and C_i represents the channel number of input features in current stage. Aided by the high resolution of f_l, the local patterns could be much reserved for subsequent extraction, but thanks to the low resolution of f_g, the exploration of non-local and object-level information is greatly benefited.

3.2 Intra-scale Propagation

Local and global features are representations of one image from two totally different views. The former focuses on fine-grained details, essential for tiny-object detection and pixel-level localization, while the latter aims at modeling object-level relations between long-range parts. Therefore, given dual-scale features f_l and f_g, we perform Intra-scale Propagation module to parallelly process them.

Local representation. For high-resolution f_l with the size of $W_i \times H_i \times C_l$, where C_l equals $\frac{C_i}{2}$, we perform 3×3 depth-wise convolution to extract local features as follows and obtain f_L:

$$f_L = \sum_{m,n}^{M,N} W(m,n) \odot f_l(i+m, j+n), \tag{1}$$

where $W(m,n)$, $(m,n) \in (-1,0,1)$ represents the convolution filters, $W(m,n)$ and $f_l(i,j)$ are both $\frac{C_i}{2}$ dimensional vectors, $\odot$ denotes element-wise product. By the power of depth-wise convolution, f_L is able to contain fine-grained local details of the input image.

Global representation. For low-resolution representation f_g with the fixed size of $\frac{W}{32} \times \frac{H}{32} \times \frac{C_i}{2}$ as illustrated in Section 3.1, we first flatten f_g to a sequence with the length of l_g, the product of $\frac{W}{32}$ and $\frac{H}{32}$, in which each element is a $\frac{C_i}{2}$ dimensional vector. By doing this, each vector in the sequence is treated as a visual token without spatial information. Here, the dependencies between different token pairs are unrelated with their spatial positions in the feature map, which is totally different with

the convolution. Then, we summarize the global information and model the object-level coupling relation via self-attention mechanism:

$$f_Q = W_Q f_g, \quad f_K = W_K f_g, \quad f_V = W_V f_g, \tag{2}$$

where W_Q, W_K, W_V denote matrixes to generate queries, keys and values respectively. By calculating the similarity between f_Q and f_V, we obtain attention weights for aggregating information from different locations of f_g. Finally, we calculate the weighted sum of attention weights and f_V, thus obtaining integrated features:

$$f_G = \mathrm{softmax}(\frac{f_Q f_K^T}{\sqrt{d}}) f_V, \tag{3}$$

where d equals $\frac{\frac{C_i}{2}}{N}$, N denotes the number of attention head that we set to 1, 2, 5, 8 for 4 stages respectively in DS-Net.

Our dual-stream architecture disentangles fine-grained and integrated features in two pathways, which significantly eliminates the ambiguity during training. Additionally, the Intra-scale Propagation module processes feature maps with dual resolutions by two domain-specifically effective mechanisms separately, extracting local and global features to the maximum extent.

3.3 Inter-scale Alignment

A delicate fusion of dual-scale representations is vital for the success of DS-Net, since they capture two different perspectives of one image. To address this, a naive idea is to upsample the low-resolution representation using bilinear interpolation, and then fuse the dual-scale representations via 1 by 1 convolution after simply concatenating them position-wisely. Such naive fusion is not good enough obviously. Furthermore, by visualizing the feature maps of dual-scale features, we observe that global features in low resolutions and local features in high-resolutions are actually misaligned (see (b) and (d) in Fig. 1. Therefore, considering the relation of two representations are not explicitly explored, it is unpersuasive to pre-define a fixed strategy to fuse them, such as concatenation, element-wise addition or production. Enlightened by [10, 26], we propose a novel co-attention-based Inter-scale Alignment module, whose scheme of Inter-scale Alignment module is shown in Fig 2. This module aims to capture the mutual correlations between each local-global token pair, and propagate information bidirectionally in a learnable and dynamic manner. Such mechanism prompts local features to adaptively explore their relations with global information, enabling themselves to be more representative and informative, and vice versa.

Given extracted local features f_L with the size of $W_i \times H_i \times \frac{C_i}{2}$ and global features f_G with the size of $l_g \times \frac{C_i}{2}$, we first flatten f_L to a sequence with the length of l_l, the product of W_i and H_i, in which each element is a $\frac{C_i}{2}$ dimensional vector. f_L and f_G are now in the same format but with different length. Then we perform co-attention on two sequences as follows:

$$\begin{aligned} Q_L &= f_L W_Q^l, \quad K_L = f_G W_K^l, \quad V_L = f_G W_V^l, \\ Q_G &= f_G W_Q^g, \quad K_G = f_G W_K^g, \quad V_G = f_G W_V^g, \end{aligned} \tag{4}$$

where the sizes of W are all $\frac{C_i}{2} \times dim$, and the dim is a hyper-parameter. Thus we have the transformed features of local and global representations. Then we calculate the similarities between every pair of f_L and f_A to obtain the corresponding attention weights:

$$W_{G\to L} = \mathrm{softmax}(\frac{Q_L K_G^T}{\sqrt{d}}), \quad W_{L\to G} = \mathrm{softmax}(\frac{Q_G K_L^T}{\sqrt{d}}). \tag{5}$$

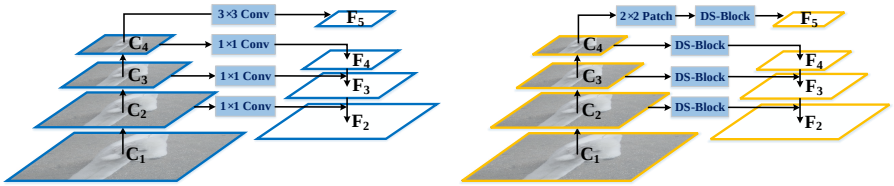
The size of $W_{G\to L}$ and $W_{L\to G}$ are $l_l \times l_g$ and $l_g \times l_l$ respectively. The non-linear function softmax is performed at the last dimension. $W_{G\to L}$ reflects the importance of different tokens in global features to the local tokens. Likewise, global features can also extract useful information from local features via $W_{L\to G}$. Instead of fixed fusion strategy that might bring constraints, what and how the information transfers are automatically determined by features themselves here. We can then obtain hybrid features as:

$$h_L = W_{G\to L} V_G, \quad h_G = W_{L\to G} V_L, \tag{6}$$

where the size of h_L and h_G are $l_l \times dim$ and $l_g \times dim$ respectively. Then we add a 1×1 convolution layer after hybrid features to further fuse the channels and reshape them to $W_i \times H_i \times \frac{C_i}{2}$ and $\frac{W}{32} \times \frac{H}{32} \times \frac{Ci}{2}$.

Such a bidirectional information flow is able to identify cross-scale relations between local and global tokens, by which dual-scale features are highly aligned and coupled with each other. After this, we could safely upsample low-resolution representation h_G, concatenate it with high-resolution h_L and perform 1 by 1 convolution for channel-wise dual-scale information fusion. At the end of the last block in stage 4, we add a fully-connected layer as the classifier to conduct classification.

3.4 Dual-stream Feature Pyramid Network



(a) FPN (b) DS-FPN

Figure 3: The architecture of DS-FPN. C_i denotes the feature maps in stages from backbone, and F_i denotes the reconstructed features for detection or segmentation.

Introducing contextual information into Feature Pyramid Networks(FPN) [18] has been explored by [39, 25]. However, previous methods often cause large extra memory and computation costs, due to their complicated architectures and utilized high resolution feature maps. Besides, Non-local contexts would miss local details, which is disastrous for tiny object detection and segmentation. Here, we apply our Dual-stream design into FPN, named Dual-stream Feature Pyramid Networks(DS-FPN), by simply adding DS-Blocks to every feature pyramid scale. In this way, DS-FPN is able to better attain non-local patterns and local details with marginal increased costs at all scales, which further enhance the performance of subsequent object detection and segmentation heads. This shows our DS-Net can not only serve as a backbone but also a general plug-in building block in many other vision architectures.

Fig 3(b) shows our DS-FPN pipeline in RetinaNet. Similar to FPN in Fig 3(a), we take image features from various scales from the backbone as input, and output corresponding refined feature maps of fixed channels number by a top-down aggregation methods. Our structure is composed of bottom-up pathways, Dual-stream lateral connections, and top-down pathways. The bottom-up and top-down pathways follow the design of FPN, but lateral connection here adopts DS-Block to process features in dual scales via Intra-scale Propagation and Inter-scale Alignment.

Specifically, we denote feature maps of bottom-up pathways as $\{C_2, C_3, C_4\}$, where C_1 is ignored due to its low semantics and high resolutions. After processed by lateral DS-Blocks and aggregated with the upsampled features from top-down pathway, we obtain final $\{F_2, F_3, F_4\}$. When extra feature outputs are needed, FPN utilizes 3×3 convolution with stride 2 to obtain F_5 from C_4, while DS-FPN adopts 2×2 patches embedding to down-sample the image and follows a DS-Block to generate dual-scale features.

4 Experiments

In this section, we first provide the ablation study to find an appropriate ratio of self-attention to convolution in Intra-scale Propagation module. Then we give the experimental results of image classification and downstream tasks including object detection and instance segmentation. Specifically, We use ImageNet-1K [7] for classification and MSCOCO 2017 [20] for object detection and instance segmentation. All experiments are conducted on 8 V100 GPUs and the throughput is tested on 1 V100 GPU. As the number of blocks we set in different stages in the architecture (see Fig. 2) is flexible, we conduct experiments on 2 models with different parameter size, denoted as DS-Net-T (Tiny) and DS-Net-S (Small). The detailed setup is shown in Table 1.

Table 1: Detailed settings of DS-Net. $Dconv$ denotes depth-wise convolution, R_i and C_i denotes the feature dimension expansion ratio and the number of channels in ith stage. The parameters indicates the complexity of DS-Net with and without Inter-scale Alignment Module.

Stage	Input size	DS-Net-T	DS-Net-S
Stage 0	224×224	4×4, 64, stride=4, padding=0	
Stage 1	56×56	$\begin{bmatrix} \mathrm{Dconv}(3\times3) \\ \mathrm{MHSA}-8 \\ \mathrm{R}_1=4, \mathrm{C}_1=64 \end{bmatrix} \times 2$	$\begin{bmatrix} \mathrm{Dconv}(3\times3) \\ \mathrm{MHSA}-8 \\ \mathrm{R}_1=4, \mathrm{C}_1=64 \end{bmatrix} \times 3$
Stage 2	28×28	$\begin{bmatrix} \mathrm{Dconv}(3\times3) \\ \mathrm{MHSA}-8 \\ \mathrm{R}_2=4, \mathrm{C}_2=128 \end{bmatrix} \times 2$	$\begin{bmatrix} \mathrm{Dconv}(3\times3) \\ \mathrm{MHSA}-8 \\ \mathrm{R}_2=4, \mathrm{C}_2=128 \end{bmatrix} \times 4$
Stage 3	14×14	$\begin{bmatrix} \mathrm{Dconv}(3\times3) \\ \mathrm{MHSA}-8 \\ \mathrm{R}_3=4, \mathrm{C}_3=320 \end{bmatrix} \times 4$	$\begin{bmatrix} \mathrm{Dconv}(3\times3) \\ \mathrm{MHSA}-8 \\ \mathrm{R}_3=4, \mathrm{C}_3=320 \end{bmatrix} \times 8$
Stage 4	7×7	$\begin{bmatrix} \mathrm{Dconv}(3\times3) \\ \mathrm{MHSA}-8 \\ \mathrm{R}_4=4, \mathrm{C}_4=512 \end{bmatrix} \times 1$	$\begin{bmatrix} \mathrm{Dconv}(3\times3) \\ \mathrm{MHSA}-8 \\ \mathrm{R}_4=4, \mathrm{C}_4=512 \end{bmatrix} \times 3$
	7×7	global average pooling, 1000-d fc, softmax	

Table 2: DS-Net-T performance on ImageNet-1k validition set with different α.

α	0	0.25	0.5	0.75	1
Top-1(%)	77.11	78	**78.06**	77.92	77.58
Top-5(%)	93.3	**94.13**	94.09	94	93.9
Params (M)	8.6	8.7	9.1	9.8	10.7
FLOPs (G)	1.573	1.578	1.592	1.615	1.647
Throughput (imgs/s)	3240	1733	1199	912	740

4.1 Ablation Study

Settings. As described in Section 3.1, we split the features into two parts in channel dimension at the beginning of each block. We conduct an experiment to explore a appropriate partition ratio. We use α to denote the proportion of channel number of f_g (global features) to the total channel number. When α equals 0, only depth-wise convolution is performed, and when α equals 1, only self-attention is performed. Noting that in this experiment, we choose DS-Net-T as the testing model, which simply concatenates f_L and f_A at the end of each block, without Inter-scale Alignment module.

Results. The results in Table 2 indicates that when α equals 0.5, DS-Net-T achieves the highest Top-1 accuracy on ImageNet-1k validation set, which, to some extent, implies that local and global information play equally important roles for visual representations. Thus, in the following experiments, we equally split features into two parts at the channel dimension to obtain dual-scale representations.

4.2 Image Classification

Settings. Image classification experiments are performed on ImageNet-1K [7], comprising 1.28M training images and 50K validation images of 1000 classes. For fair comparison with other works, we follow the training settings in DeiT. We train our model for 300 epochs by AdamW optimizer. The initial learning rate is set to 1e-3 and scheduled by the cosine strategy.

Results. We report the results in Table 3, where DS-Net and DS-Net* denote our proposed networks with and without Inter-scale Alignment module respectively. For DS-Net, only concatenation operation is used to fuse the local and global features. DS-Net* is the version with Inter-scale Alignment module. With comparable parameters and complexity, our proposed DS-Net consistently outperforms the DeiT [33] baseline by significant margins, which is, specifically, 5.86% improvement for tiny model, 2% improvement for small model. The efficacy of Inter-scale Alignment module is also clearly indicated in Table 3, which gains further 0.98% and 0.4% improvement compared to the simple version of DS-Net-T and DS-Net-S. Notably, without bells and whistles, DS-Net-T*

Table 3: Comparison with the accuracy of other state-of-art methods on ImageNet-1k validition set. The input images are reshape to 224 $\times$ 224 resolution. DS-Net* represents the corresponding DS-Net version with Inter-scale Alignment module.

Method	Params (M)	Flops (G)	Throughput (Images/s)	Top-1 (%)
ConvNet				
ResNet-18 [13]	11.8	2	-	69.9
ResNet-50 [13]	25.6	4.1	-	74.2
ResNet-101 [13]	4.5	7.8	-	77.4
Transformer / Hybrid				
Deit-Tiny [33]	6	-	2536	72.2
CPVT-Ti [6]	6	-	-	72.4
T2T-ViT-12 [38]	6.9	-	-	76.5
DS-Net-T (ours)	9.09	1.5	1199	78.06
DS-Net-T* (ours)	10.5	1.8	1034	**79.04**
Deit-Small/16 [33]	22.1	4.6	940	79.9
CrossViT-S [3]	26.7	5.6	-	81.0
T2T-ViT-14 [38]	22	5.2	-	81.5
TNT-S [11]	23.8	5.2	-	81.3
PVT-Small [35]	24.5	3.8	820	79.8
CPVT-Small-GAP [6]	23	4.6	817	81.5
Swin-T [22]	29	4.5	766	81.3
Twins-SVT-S [5]	24	2.8	1059	81.7
Conformer-T [27]	23.5	5.0	610	81.3
DS-Net-S (ours)	19.7	3.0	582	81.9
DS-Net-S* (ours)	23	3.5	510	**82.3**

Table 4: Comparison with the performance for object detection and instance segmentation on the MSCOCO minival set of other state-of-art methods.

Method	Backbone	Params (M)	AP^b	AP^b_S	AP^b_M	AP^b_L	AP^s	AP^s_S	AP^s_M	AP^s_L
RetinaNet [19]	ResNet-50 [13]	37.7	36.3	19.3	40.0	48.8	-	-	-	-
	Swin-T [22]	38.5	41.5	25.1	44.9	55.5	-	-	-	-
	DS-Net-S* (ours)	33.2	**42.7**	**26.8**	**46.3**	**56.7**	-	-	-	-
Mask R-CNN [12]	ResNet-50 [13]	44.2	38.2	21.9	40.9	49.5	34.7	18.3	37.4	47.2
	Swin-T [22]	48	43.7	**28.5**	47	57.3	39.8	**24.2**	43.1	54.6
	DS-Net-S* (ours)	43.2	**44.3**	28.3	**47.7**	**58.8**	**40.2**	24	**43.4**	**54.7**

DS-Net-S* also outperform other state-of-the-art methods, including CNN, vision transformer and some hybrid architectures.

4.3 Object Detection and instance segmentation

Settings. Experiments in this part are conducted on the challenging MSCOCO 2017 [20], containing 118K training images and 5K validation images. Specifically, the backbone is initialized by pretraind weights from ImageNet-1K, and other layers adopt Xavier. We evaluate our DS-Net on typical detectors: RetinaNet and Mask R-CNN [12] and Cascade Mask R-CNN [1, 12]. As the standard 1$\times$ schedule(12 epochs), we adopt AdamW optimizer with initial learning rate of 1e-4, decayed by 0.1 at epoch 8 and 11. We set stochastic drop path regularization of 0.1 and weight decay of 0.05. Following previous methods, training and testing images are resized to 800$\times$1333.

Results. The results are shown in Table 4. For object detection, DS-Net-S* achieves significant improvement compared to ResNet-50 backbone [19], and outperforms the previous state-of-the-art Swin-T [22] by 1.2% in terms of AP^{bbox}. For instance segmentation, DS-Net-S* achieves 40.2% in terms of AP^{segm}, which surpasses the ResNet-50 and Swin-T by by 5.5% and 0.4%.

4.4 Dual-stream Feature Pyramid Networks

Settings. Experiments of DS-FPN are implemented on DS-Net-T and Swin Transformer [22] for object detection and instance segmentation on MSCOCO 2017 [20], by replacing the original FPN with DS-FPN. The training setting and scheme are the same as those mentioned above, which is $1\times$ schedule(12 epochs) and AdamW optimizer with initial learning rate of 1e-4. Likewise, the backbone is initialized with pretrained weights, and DS-FPN is trained from scratch with newly added heads.

Ablation Study. In Table 5, we report results of several different designs combining DS-Blocks with FPN on Retinanet, with DS-Net-T backbone. Four inserted positions of DS-Blocks have been experimented, where "Last" means replacing the last 3×3 convolution layer, and "Lateral" equips them on lateral connections. More specifically, "Lateral$_{rev}$" moves the inner channel-alignment 1×1 convolution layer from the last place to the first, and "Lateral$_{extra}$" adds DS-Blocks also on extra feature outputs. The results show that "Lateral$_{extra}$" has the best performance. Therefore, we set "Lateral$_{extra}$" as our default, which is 1.4% higher than FPN with marginal memory costs.

Comparison with FPN. As reported in Table 6, for object detection, DS-Net-T and Swin Transformer with DS-FPN gain 1.4% and 0.8% mAP improvement, respectively. For instance segmentation, DS-FPN helps to gain 1.2% and 1.4% AP^{segm} improvement for DS-Net-T and Swin Transformer, respectively. The results significantly prove the generality and effectiveness our Dual-stream design.

Table 5: Performance of DS-Blocks' different inserted positions on FPN of RetinaNet.

Inserted Position	None	Last	Lateral	Lateral$_{rev}$	Lateral$_{extra}$
AP^{bbox}	39	40.2	40.3	40.2	**40.4**
Params (M)	18.83	19.59	23.72	21.36	24.43

Table 6: Comparison on the performance of DS-FPN and FPN for object detection and instance segmentation on MSCOCO 2017.

Method	Backbone	Neck	Params(M)	AP^b	AP^b_S	AP^b_M	AP^b_L	AP^s	AP^s_S	AP^s_M	AP^s_L
RetinaNet [19]	DS-Net-T	FPN	18.83	39	23.3	42.6	51.3	-	-	-	-
		DS-FPN	24.43	**40.4**	**26.1**	**44.2**	**52.3**	-	-	-	-
	Swin-T [22]	FPN	38.46	41.5	26.4	45.1	**55.7**	-	-	-	-
		DS-FPN	48.76	**42.3**	**27.7**	**45.9**	55.0	-	-	-	-
Mask R-CNN [12]	DS-Net-T	FPN	28.75	40.1	24.9	43.4	52.1	37.2	21.4	40.4	50
		DS-FPN	33.69	**41.3**	**25.8**	**44.6**	**53**	**37.9**	**22.1**	**41.1**	**50.4**
	Swin-T [22]	FPN	47.79	42.5	26.3	45.9	56.1	39.2	22.8	42.4	53.8
		DS-FPN	57.83	**44.2**	**27.6**	**47.6**	**58.0**	**40.6**	**23.9**	**43.9**	**55.2**

5 Conclusion

In this paper, we propose a generic Dual-stream Network, called DS-Net, which disentangles local and global features by generating two representations with different resolutions. We present Intra-scale Propagation module and Inter-scale Alignment module to combine the merits of convolution and self-attention mechanisms, and identify cross-scale relations between local and global tokens. Besides, for downstream tasks, we design a Dual-stream Feature Pyramid Network (DS-FPN) to introduce contextual information to feature pyramid for further refinement with marginal costs. With the outstanding performance on image classification and dense downstream tasks including object detection and instance segmentation, our proposed DS-Net has shown its promising potential in vision tasks.

Limitations: DS-Net is effective at image classification and downstream task including object detection and instance segmentation. However, several reshaping operations in Inter-scale Alignment module slightly slow down the training and inference speed. We expect future works to successfully address it for DS-Net.

Negative social impact: This work does not have a direct negative social impact. Despite of this, we should be aware of the power of neural networks and prevent them from being used for harmful applications.

References

[1] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *IEEE CVPR*, pages 6154–6162, 2018.

[2] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. *arXiv:2005.12872*, 2020.

[3] C.-F. Chen, Q. Fan, and R. Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. *arXiv:2103.14899*, 2021.

[4] F. Chollet. Xception: Deep learning with depthwise separable convolutions. *arXiv:1610.02357*, 2017.

[5] X. Chu, Z. Tian, Y. Wang, B. Zhang, H. Ren, and X. W. et al. Twins: Revisiting the design of spatial attention in vision transformers. *arXiv:2103.00112*, 2021.

[6] X. Chu, Z. Tian, B. Zhang, X. Wang, X. Wei, and H. Xia. Conditional positional encodings for vision transformers. In *IEEE CVPR*, 2021.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE CVPR*, pages 248–255, 2009.

[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, and T. U. et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[9] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. Centernet: Object detection with keypoint triplets. In *IEEE CVPR*, 2019.

[10] P. Gao, Z. Jiang, H. You, P. Lu, S. C. H. Hoi, X. Wang, and H. Li. Dynamic fusion with intra- and inter-modality attention flow for visual question answering. In *IEEE CVPR*, pages 6632–6641, 2019.

[11] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang. Transformer in transformer. *arXiv:2103.00112*, 2021.

[12] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. In *IEEE ICCV*, pages 2980–2988, 2017.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE CVPR*, pages 770–778, 2016.

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*, 2017.

[15] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *IEEE CVPR)*, 2018.

[16] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 765–781, 2018.

[17] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[18] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *IEEE CVPR*, pages 936–944, 2017.

[19] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *IEEE ICCV*, pages 2999–3007, 2017.

[20] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. In *ECCV*, pages 21–37, 2016.

[22] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, and Z. Z. et al. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv:2103.14030*, 2021.

[23] J. Long, E. Shelhamer, and T. Darrell. Image segmentation fully convolutional networks for semantic segmentation. In *IEEE CVPR*, 2015.

[24] M. Mao, Y. Tian, B. Zhang, Q. Ye, W. Liu, G. Guo, and D. Doermann. iffDetector: Inference-aware feature filtering for object detection. In *arXiv preprint arXiv:2006.12708*.

[25] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin. Libra r-cnn: Towards balanced learning for object detection. In *IEEE CVPR*, 2019.

[26] G. Peng, H. You, Z. Zhang, X. Wang, and H. Li. Multi-modality latent interaction network for visual question answering. In *IEEE ICCV*, pages 5824–5834, 2019.

[27] Z. Peng, W. Huang, S. Gu, L. Xie, Y. Wang, J. Jiao, and Q. Ye. Conformer: Local features coupling global representations for visual recognition. *arXiv:2105.03889*, 2021.

[28] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *IEEE CVPR*, pages 779–788, 2016.

[29] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015.

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2015.

[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE CVPR*, pages 1–9, 2015.

[32] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.

[33] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers distillation through attention. *arXiv:2012.12877*, 2020.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, and A. N. G. et al. Attention is all you need. In *NeurIPS*, 2017.

[35] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, and D. L. et al. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv:2102.12122*, 2021.

[36] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, and L. Y. et al. Cvt: Introducing convolutions to vision transformers. *arXiv:2103.15808*, 2021.

[37] H. Yan, Z. Li, W. Li, C. Wang, M. Wu, and C. Zhang. Contnet: Why not use convolution and transformer at the same time? *arXiv:2104.13497*, 2021.

[38] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, and Z. J. et al. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv:2101.11986*, 2021.

[39] D. Zhang, H. Zhang, J. Tang, M. Wang, X. Hua, and Q. Sun. Feature pyramid transformer. In *ECCV*, 2020.

[40] X. Zhang, F. Wan, C. Liu, R. Ji, and Q. Ye. FreeAnchor: Learning to match anchors for visual object detection. In *NeurIPS*, 2019.

[41] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *IEEE CVPR*, pages 6848–6856, 2018.

[42] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, and Y. W. et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *IEEE CVPR*, 2021.

[43] D. Zhou, B. Kang, X. Jin, L. Yang, X. Lian, and Z. J. et al. Deepvit: Towards deeper vision transformer. In *IEEE CVPR*, 2021.

[44] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai. Asymmetric non-local neural networks for semantic segmentation. In *IEEE ICCV*, pages 593–602, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The instructions is shown at section 4. The code is provided in supplementary materials.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We clarify corresponding experimental settings before the results.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Training the large neural networks in this paper is very expensive. Given our limited computational resources, we are unfortunately unable to provide error bars for our experiments.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] All experiments are conducted on 8 V100 GPUs and the throughput is tested on 1 V100 GPU as stated at section 4.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]