





Pyramid Swin Transformer: Different-Size Windows Swin Transformer for Image Classification and Object Detection

Chenyu Wang^{1,2}^a, Toshio Endo¹^b, Takahiro Hirofuchi²^c and Tsutomu Ikegami²^d

¹*Tokyo Institute of Technology, Tokyo, Japan*

²*National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan*

Keywords: Swin Transformer, Object Detection, Image Classification, Feature Pyramid Network, Multiscale.

Abstract: We present the Pyramid Swin Transformer for object detection and image classification, by taking advantage of more shift window operations, smaller and more different size windows. We also add a Feature Pyramid Network for object detection, which produces excellent results. This architecture is implemented in four stages, containing different size window layers. We test our architecture on ImageNet classification and COCO detection. Pyramid Swin Transformer achieves 85.4% accuracy on ImageNet classification and 54.3 box AP on COCO.

1 INTRODUCTION

Both image classification and object detection are critical tasks in computer vision, and they are used to identify objects for categorization, such as humans, animals, fruits, or some buildings. Object detection will be more complicated than image classifications since it must recognize the positions of the objects and produce marks. Currently, object detection is used for security, medical, self-driving cars, identity identification, and other purposes. It has experienced an exponential expansion in recent years, along with the rapid development of new tools and procedures. As convolutional neural networks have been utilized successfully in computer vision, research on convolutional neural networks has flourished. They have been used for various computer vision tasks, including image classification and object detection. In 2012, Alexnet(Krizhevsky et al., 2017) achieved a new accuracy record in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC)(Russakovsky et al., 2015).

After Alexnet(Krizhevsky et al., 2017), research on CNN has shown a blowout outbreak, and research on various backbone networks such as GoogLeNet (Szegedy et al., 2015), VGG (Simonyan and Zisserman, 2014), and ResNet (He et al., 2016) have followed one after another. At the same time, research

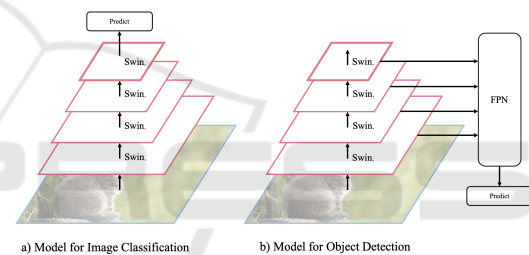





Figure 1: Pyramid Swin Transformer. a) model is used for image classification, where each layer is feature sampled with Swin Transformer, and b) model is used for object detection. It adds a feature pyramid network to a) model.


on the object detection framework is also developing, such as Faster R-CNN(Ren et al., 2015) and Mask R-CNN(He et al., 2017), which are two-stage detectors. A module presents some candidates, which the network classifies as objects or backgrounds. Two-stage detectors can be slower but more accurate than single-stage detectors, such as YOLO (Redmon et al., 2016), Single Shot MultiBox Detector (SSD) (Liu et al., 2016), which are single-stage detectors.

More recently, convolutional neural networks (CNN) have been outperformed by Vision Transformers (Vit), which have shown potential and have been modified for usage in various vision applications.(Arnab et al., 2021; Carion et al., 2020; Beal et al., 2020; Strudel et al., 2021; Wang et al., 2021). The Transformer (Vaswani et al., 2017) model is a relatively new deep learning model. However, it has been widely researched and applied in natural language processing (NLP) and computer vision (CV). The Transformer was initially introduced as a

^a <https://orcid.org/0000-0001-8770-3275>

^b <https://orcid.org/0000-0001-7297-6211>

^c <https://orcid.org/0000-0002-1253-6625>

^d <https://orcid.org/0000-0003-2977-6390>

machine translation sequence-to-sequence (Sutskever et al., 2014) concept. Later studies have shown that Transformer-based pre-trained models (PTMs)(Qiu et al., 2020) can achieve cutting-edge performance on various tasks. As a result, The Transformer has become the most preferred design in NLP, particularly for PTMs. In recent research(Dosovitskiy et al., 2020), the Transformer model has also performed well in CV tasks. Naturally, Transformer recently has been utilized in CV(Dosovitskiy et al., 2020; Liu et al., 2021) and audio processing(Chen et al., 2021; Dong et al., 2018).

Our Pyramid Swin Transformer is an improved version of the Swin Transformer(Liu et al., 2021), and we propose two models, one for image classification and another for object detection. We improve the original Swin Transformer by using smaller, more different-size windows and more shift window operations in order to achieve a better detection effect, which certainly enhances the detection effect. Compared to Swin Transformer, our Pyramid Swin Transformer uses windows of varying sizes on a unified scale to execute multiple window multi-head self-attention computation, as shown in Figure 2. This slightly increases the amount of calculation, but it better solves the problem of information interaction between windows and windows.

2 RELATED WORK

The use of the Transformer for computer vision is arguably an essential attempt in the history of computer vision. Transformer-like frameworks are what allow us to get rid of CNN and have better globalization. With more in-depth research, the model will also become more applicable to computer vision, capable of performing more tasks quickly and accurately. The model will also become more useful in computer vision and can perform more quickly and accurately. There is no doubt that Swin Transformer(Liu et al., 2021) is a promising improvement.

2.1 Feature Pyramid Network

The Feature Pyramid Network (FPN)(Lin et al., 2017) is a high-accuracy and fast feature extractor designed for this pyramid concept. It can replace detector feature extractors like Faster R-CNN(Ren et al., 2015) and generates many feature layers (multi-scale feature maps) with higher quality information for object detection than traditional feature pyramids. FPN is made up of a bottom-up and a top-down pathway. The bottom-up route is the standard convo-

lutional network for feature extraction. The spatial resolution degrades as they ascend, and the semantic value of each layer grows as more high-level structures are recognized. FPN offers a top-down approach to building higher-resolution layers from a semantic-rich layer. While the reconstructed layers are semantically strong, the positions of objects after all the down-sampling and up-sampling are not exact. They improve the detector's prediction by adding lateral links between reconstructed layers and the associated feature map. FPN has been widely employed in several frameworks for object detection(Ronneberger et al., 2015; Zhang et al., 2018; Peng et al., 2018) and semantic segmentation(Liu et al., 2018) because of its excellent results and practicality, and all of them have obtained successful outcomes. FPN has been proven to significantly improve object detection accuracy with a modest increase in processing cost.

2.2 Swin Transformer

The first part of the name Swin Transformer(Liu et al., 2021) is derived from Shifted Windows, which is also the main feature of Swin Transformer. The research community is not new to the idea of shifted windows. As a result of its high efficiency, it is one of the CNN aspects that has helped the network succeed in the field of computer vision. However, it had not been used in Transformers before. The original intention of Swin Transformer's authors was to make Vision Transformer-like a convolution neural network, which can also be divided into several blocks for cascading feature extraction, thus leading to the concept of multi-scale for the proposed features. The standard Transformer(Dosovitskiy et al., 2020) has some challenges when applied directly to the vision domain. The difficulty comes mainly from two aspects, namely different scales and the large resolution of the image. The first problem is scale, but this phenomenon does not exist in natural language processing. To solve the problem of sequence length, researchers have made a series of attempts, including using the subsequent feature map as input to the Transformer or breaking the image into multiple patches to reduce the resolution of the image as well as dividing the images into small windows and then doing self-attention(Vaswani et al., 2017) computation in the windows.

Since the self-attention is computed within the window, its computational complexity grows linearly rather than squarely with the image size, giving Swin Transformer the ability to pre-train models at particularly high resolutions, the hierarchical structure has the advantage of being flexible enough to provide in-

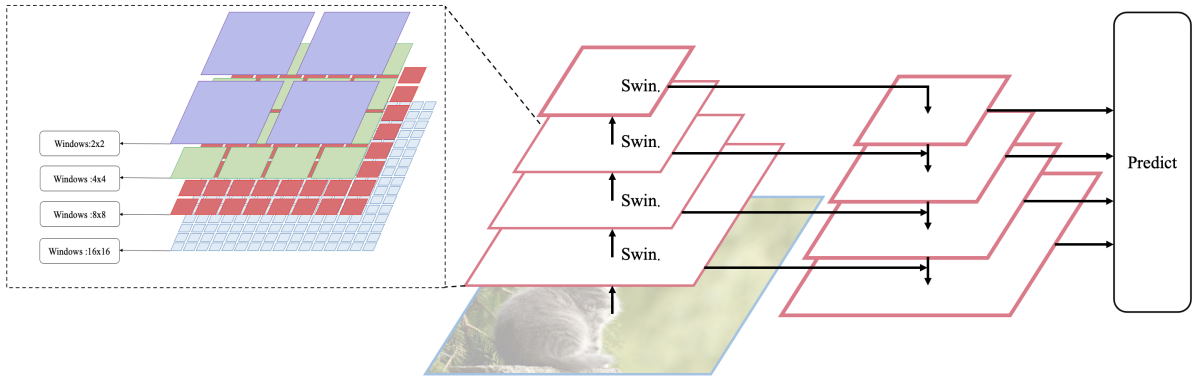


Figure 2: Pyramid Swin Transformer. The part on the right is the overall Pyramid Swin Transformer, where Swin Transformer computes each layer, and the part on the far right is the part of the Feature Pyramids Network. The part shown on the left is our Pyramid Swin Transformer, where we use a different size window to compute multi-head self-attention.

formation at different scales. Following the original Vision Transformer, the Swin Transformer is likely one of the most exciting pieces of research. The Swin Transformer resolved the original ViT’s problems by using hierarchical feature maps and shifting window multi-head self-attention. The Swin Transformer is now widely applied as a backbone architecture in various vision tasks, such as image classification and object detection.

Swin Transformer does a wonderful job of resolving the multi-scale and computational complexity issues in ViT (Dosovitskiy et al., 2020), but it also brings a few new issues due to the usage of window multi-head self-attention, which makes the windows independent of one another. Although the authors incorporated shift-window multi-head self-attention, which enables specific windows to be connected, there is still a lack of information interaction between some windows in large-scale dimensions. Therefore, we present the Pyramid Swin Transformer in order to look forward to solving the communication problem between windows. As a result, it seems to deal with the problem at some extent.

3 METHOD

Recent developments in vision Transformer backbone designs are mostly concerned with attention operator advancements. A new topology design can add a new dimension to ViTs, allowing for even more powerful vision expressivity. One of the best is the Swin Transformer (Liu et al., 2021), on which our research is based. In this section, we will go into the structure of our Pyramid Swin Transformer.

3.1 Architecture

Our architecture is mainly based on the Swin Transformer (Liu et al., 2021), as shown in Figure 2, our architecture for object detection adds a feature pyramid part and adjusts the original Swin Transformer. We use a hierarchical network, where the first stage size is the largest one (64×64). In this stage, we divide the feature map into four types (16×16 , 8×8 , 4×4 , 2×2 windows), which corresponding window sizes are 4×4 , 8×8 , 16×16 and 32×32 . In the window size of 32×32 , the length of self-attention is 1024, which will bring much computation, so this size of the window we only use in stage 1, other stages will never utilize this size window.

Each layer consists of two steps, one for window multi-head self-attention and the next for shift window multi-head self-attention, each layer is the same and includes two computations of multi-head self-attention, which is also the same in Swin Transformer (Liu et al., 2021), as shown in Figure 3, except that we split it into smaller blocks, and the number of different-size windows in each layer is the hierarchical progression from more to less, which is more conducive to global connection. When computing shift window multi-head self-attention, we only compute shift window multi-head self-attention once in each layer in order to decrease the computation. Each stage of the last layer, except the fourth stage will have a 2×2 window, increasing window-to-window information interaction. As a result, increasing global relevance.

The overall architecture is shown in Table 1. The size of our input image is 256×256 , we call it Pyramid Swin-R, and the first stage has 4 layers, the second stage has 3 layers, the third stage has 2 layers, and the last stage has 2 layers. We also implement another framework, the Pyramid Swin-T, which differs from

the Pyramid Swin-L only in the number of channels and layers. The details are as follows:

- Pyramid Swin-R: $C=96$, $\text{layer}=\{4,3,2,2\}$
- Pyramid Swin-L: $C=192$, $\text{layer}=\{4,3,3,2\}$

where C is the layer channel number in the first stage. Each layer consists of two sub-multi-head self-attention calculations.

We utilize different window sizes in each layer, and the next layer complements the previous one. The problem with Swin Transformer was insufficient information interaction between windows and windows at the low semantic level, and our improvements have enhanced the information interaction between separate windows. The main concept behind the Pyramid Swin Transformer is to keep adding windows of various sizes to a uniform scale to improve their direct information contact.

Supposing each window contains $2^i \times 2^i$ window sizes, on an image of size $h \times w$ feature map, the computational complexity of a global multi-head self-attention module and a window-based one is as follows:

$$\Omega(MSA) = 4hwC^2 + 2(hw)^2C$$

$$\Omega(W - MSA) = 4hwC^2 + (2)^{2i+1}hwC$$

Where C is a channel and the former is quadratic to feature map size $h \times w$, and the latter depends mainly on the size of i , which $i \in \{0, 5\}$. In our design, because the computation is too large when $i = 5$, we try to minimize the case of $i = 5$ in the whole framework. Actually, we only use $i = 5$ once at the first layer of the first stage. While window multi-head self-attention is scalable for $h \times w$, global self-attention computation is typically costly. Therefore, window self-attention has excellent potential for lowering computation requirements.

For self-attention computation, we follow (Raffel et al., 2020; Hu et al., 2019; Liu et al., 2021) by including a relative position bias $\beta \in \mathbb{R}^{M^2 \times M^2}$ to each head:

$$\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T / \sqrt{d} + \beta)V,$$

where $Q, K, V \in \mathbb{R}^{M^2 \times d}$ are the *query*, *key* and *value* matrices, d is the *query* and *key* dimension and M^2 is the window size.

3.2 Pyramid Swin for Object Detection

We will introduce how to apply the Pyramid Swin Transformer backbone for object detection. Pyramid Swin Transformer's hierarchical structure generates multiscale feature maps in four phases. We integrate

neatly into Feature Pyramid Networks (FPN) for object detection applications, as shown in figure 2. Our Pyramid Swin Transformer creates semantically robust feature maps at all scales using a top-down pyramid with lateral connections in FPN. By using FPN with the Pyramid Swin Transformer backbone, we apply it to different detection architectures. In Feature Pyramid Network, we use pixel shuffle for implementation to upsample feature maps of small size, which in our experiments can achieve better results than the normal pooling method. We have always set the feature pyramid's channel to 96, and it correlates to the Swin Transformer's portion on the left side of the pyramid.

Object detection often uses training inputs with a range of input sizes than ImageNet classification, where the input image is a crop with a set resolution (such as 224×224). We initialize the parameters from the ImageNet pre-training weights to respond to the location embedding with a 256×256 input size and then interpolate them to their respective sizes for object detection training.

4 EXPERIMENT

We conduct experiments on ImageNet-1K image classification (Deng et al., 2009), COCO object detection (Lin et al., 2014). In the following sections, we will compare the suggested Pyramid Swin Transformer architecture to the prior state-of-the-art on two tasks.

4.1 Image Classification on ImageNet

Settings. To be fair, we benchmark the proposed Pyramid Swin Transformer on ImageNet-1K (Deng et al., 2009), which contains 1.28M training images and 50K validation images from 1,000 classes. On a single crop, the top-1 accuracy is reported. Swin Transformer is used in our training methods (Liu et al., 2021). ImageNet-1K training. This setting mostly follows (Touvron et al., 2021). We employ an AdamW (Kingma and Ba, 2014) optimizer for 300 epochs using a cosine decay learning rate scheduler, as same as the Swin Transformer (Liu et al., 2021). We include most of the augmentation and regularization strategies of (Touvron et al., 2021) in training, except for repeated augmentation (Hoffer et al., 2020) and EMA (Polyak and Juditsky, 1992). Note that this contrasts the situation where consistent augmentation is essential to maintain ViT training (Dosovitskiy et al., 2020).

Image Classification on ImageNet. Table 2 shows the results of our Pyramid Swin Transformer and

Table 1: Pyramid Swin Transformer Detailed architecture specifications. Input image size is 256×256 .

Pyramid Swin-R	Output Size	Layers	Channel	Windows	Window size	Heads
Stage 1	64^2	4	96	$16^2, 8^2, 4^2, 2^2$	$4^2, 8^2, 16^2, 32^2$	3
Stage 2	32^2	3	192	$8^2, 4^2, 2^2$	$4^2, 8^2, 16^2$	6
Stage 3	16^2	2	384	$4^2, 2^2$	$4^2, 8^2$	12
Stage 4	8^2	2	768	$2^2, 1^2$	$4^2, 8^2$	24
Pyramid Swin-L	Output Size	Layers	Channel	Windows	Window size	Heads
Stage 1	64^2	4	192	$16^2, 8^2, 4^2, 2^2$	$4^2, 8^2, 16^2, 32^2$	3
Stage 2	32^2	3	384	$8^2, 4^2, 2^2$	$4^2, 8^2, 16^2$	6
Stage 3	16^2	3	768	$8^2, 4^2, 2^2$	$2^2, 4^2, 8^2$	12
Stage 4	8^2	2	1536	$2^2, 1^2$	$4^2, 8^2$	24

Table 2: Test Environment.

CPU	Intel(R) Xeon(R) Silver 4110
Memory	16G
GPU	NVIDIA Tesla V100 PCIe
GPU Memory	16G
Pytorch	1.7.1
CUDA	11.6
OS	Ubuntu 18.04

state-of-the-art CNNs and Transformers. Based on computation, the models are divided into categories. Compared with the state-of-the-art Convolution Nets and Vision transformer models such as RegNet (Radosavovic et al., 2020), EfficientNet (Tan and Le, 2019), CoAtNet (Dai et al., 2021), ViT (Dosovitskiy et al., 2020), DeiT (Touvron et al., 2021), MVit (Fan et al., 2021), Swin (Liu et al., 2021) and SwinV2 (Liu et al., 2022), our Pyramid Swin Transformer achieves slightly better accuracy. All details are shown in Table 3.

Comparison on Imagenet. Our design outperforms several CNN systems even when we utilize a regular model (Pyramid Swin-R). In image classification, our design has no evident advantages over Transformer systems. Compared to Swin Transformer (Liu et al., 2021), our improved Pyramid Swin Transformer has greater accuracy than Swin Transformer while using fewer FLOPs and parameters. For example, Pyramid Swin-R (84.6%) achieved the same result as SwinV2-B (Liu et al., 2022). This shows that the overall architecture of Swin Transformer has a lot of limitations, and the immediate increase in the number of parameters does not yield good results. On the regular-size model, Pyramid Swin-R (84.6%) improves +0.1% over Swin-B but with fewer FLOPs and parameters. On the large-size model, Pyramid Swin-

Table 3: Comparison with previously reported ImageNet-1K work. We make pretrain on ImageNet-1K. Pyramid Swin is trained for 300 epochs without any external data or models.

Method	Resolution	Params	FLOPs	Top-1 Acc.
RegNetY-4G	224^2	21M	4G	80.0
RegNetY-8G	224^2	39M	8G	81.7
RegNetY-16G	224^2	84M	16G	82.9
EfficientNet-B1	240^2	8M	1G	78.8
EfficientNet-B2	260^2	9M	1G	79.8
EfficientNet-B3	300^2	12M	2G	81.6
EfficientNet-B4	380^2	19M	4G	82.9
EfficientNet-B5	456^2	30M	10G	83.6
EfficientNet-B6	528^2	43M	19G	84.0
EfficientNet-B7	600^2	66M	37G	84.4
CoAtNet-0	224^2	25M	4G	81.6
CoAtNet-1	224^2	42M	8G	83.3
CoAtNet-2	224^2	75M	16G	84.1
CoAtNet-3	224^2	168M	35G	84.6
ViT-B/16	384^2	86M	55G	77.9
ViT-L/16	384^2	307M	191G	76.5
DeiT-S	224^2	22M	5G	79.8
DeiT-B	224^2	86M	18G	81.8
DeiT-B	384^2	86M	55G	83.1
MViT-B-16	224^2	37M	8G	83.0
MViT-B-24	224^2	72M	15G	84.0
MViT-B-24	320^2	73M	33G	84.8
Swin-T	224^2	28M	5G	81.3
Swin-S	224^2	50M	9G	83.0
Swin-B	224^2	88M	15G	83.5
Swin-B	384^2	88M	47G	84.5
SwinV2-T	256^2	28M	7G	82.8
SwinV2-S	266^2	50M	13G	84.1
SwinV2-B	256^2	88M	22G	84.6
P. Swin-R	256^2	77M	18G	84.6
P. Swin-L	256^2	164M	39G	85.4

L (85.4%) improves +0.8% over SwinV2-B (84.6%). Our Pyramid Swin Transformer is only more accurate than Swin-B +0.1% with an equal size model. However, Swin-B utilizes a higher resolution. We get the same accuracy as SwinV2-B with the same image size, but our computation is fewer. Compared with MVit (320×320), our large model(Pyramid Swin-L) has a higher accuracy rate, but the amount of computation is also greatly increased. Compared with MVit (224×224), our regular model(Pyramid Swin-R) has a +0.6% accuracy. The effect of our architecture on image classification is not so obvious, and the main contribution is to reduce the same amount of computation while maintaining accuracy.

4.2 Object Detection on COCO

Settings. We conduct object detection experiments on the Microsoft COCO(Lin et al., 2014) dataset. An ablation study is conducted using the validation set, and test-dev is used to report on a system-level comparison. We use standard Mask R-CNN (He et al., 2017) and Cascade Mask R-CNN (Cai and Vasconcelos, 2018) detection frameworks implemented in Detectron. The backbone networks of the objects we compared are Resnet(He et al., 2016), ResNet(Xie et al., 2017), PVT-S(Wang et al., 2021), ViL-S-RPB(Zhang et al., 2021) and Swin(Liu et al., 2021). For a fair comparison, we follow the same way as Swin Transformer (Liu et al., 2021). For these four frameworks, we utilize the same settings: multi-scale training (Carion et al., 2020; Sun et al., 2021). For Pyramid Swin, we take the backbone pre-trained from Imagenet-1K. The input sizes are set as $[64, 32, 16, 8]$ for multi-scale four stages, consistent with the self-attention size used in Imagenet-1K pre-training.

With Mask R-CNN. On the regular size model, our Pyramid Swin achieves the highest accuracy when we utilize the framework of Mask R-CNN. Pyramid Swin-R 50.3 box AP improves +1.8 box AP over Swin-B(Liu et al., 2021) with fewer FLOPs and parameters. Compared to ViL-B-RPB, our Pyramid Swin has an advantage, with +0.7 box AP improvement. On the large-size model Pyramid Swin-L achieves 51.6 box AP, improving +3.1 box AP over Swin-B with far more FLOPs and parameters.

With Cascade Mask R-CNN. On the regular size model, our Swin also achieves the highest accuracy when we utilize the framework of Cascade Mask R-CNN. Pyramid Swin-R gets 53.6 box AP improving +1.7 box AP over Swin-B with fewer FLOPs and parameters. Our Pyramid Swin-L achieves 54.3 box AP for the large-size model, improving +2.4 box AP over Swin-B. Our Pyramid Swin Transformer is less effective

Table 4: Results on COCO object detection.C.Mask indicates Cascade Mask R-CNN, R.PointsV2 indicates Rep-PointsV2.

a) Mask R-CNN				
Model	AP^{box}	AP^{mask}	FLOPs	Params
Res50	41.0	37.1	260G	44M
Res101	42.8	38.5	336G	63M
X101-64	44.4	39.7	493G	101M
PVT-S	43.0	39.9	245G	44M
PVT-M	44.2	40.5	302G	64M
PVT-L	44.5	40.7	364G	81M
ViL-S-RPB	47.1	42.1	277 G	45M
ViL-M-RPB	48.9	44.2	352G	60M
ViL-B-RPB	49.6	44.5	384G	76M
Swin-T	46.0	41.6	264G	48M
Swin-S	48.5	43.3	354G	69M
Swin-B	48.5	43.4	496G	107M
P. Swin-R	50.3	44.8	463G	94M
P. Swin-L	51.6	45.3	1014G	193M
b) Cascade Mask R-CNN				
Model	AP^{box}	AP^{mask}	FLOPs	Params
Res50	46.3	40.1	739G	82M
Res101	47.7	40.8	819G	101M
Swin-T	50.5	43.7	745G	86M
Swin-S	51.8	44.7	838G	107M
Swin-B	51.9	45.0	982G	145M
P. Swin-R	53.6	46.4	902G	136M
P. Swin-L	54.3	47.1	1867G	273M

in image classification than in object detection because we did not improve Swin’s original framework significantly, improving the effect on image classification a little, not obvious. In contrast, a feature pyramid network is added for object detection, obtaining more significant results. As seen from Swin, from Swin-S to Swin-B, the number of parameters and FLOPs are also increased, but there is no significant effect. We can conclude that if computing does not significantly rise, it may be the limit of the Swin framework. We have addressed this weakness by slightly increasing computation while still obtaining an acceptable outcome.

5 CONCLUSION

This time we publish an improved version of the Swin Transformer, the Pyramid Swin Transformer, where we use windows of different sizes to perform window multiple multi-head self-attention operations on

the same scale, improving the Swin Transformer. For the Pyramid Swin Transformer, we created two models for image classification and object detection. For image classification, our Pyramid Swin-R achieves the same results as SwinV2-B(Liu et al., 2022) on the Imagenet-1k test. At the same time, the Pyramid Swin-L model outperforms the original framework by +0.6%, and we achieve better results for object detection. When we use the Mask R-CNN framework, Pyramid Swin-R achieves 50.3 box AP and 44.8 mask AP, Pyramid Swin-L achieves 51.6 box AP and 45.3 mask AP, and when we use the Cascade Mask R-CNN framework, our Pyramid Swin-R gains 53.6 box AP and 46.4 mask AP. Pyramid Swin-L achieves 54.3 box AP and mask AP, improving Swin Transformer box AP and mask AP significantly. In the future, we aim to do some speed tests and develop some lighter architectures for more in-depth comparative testing with existing architectures. We will add Semantic Segmentation on ADE20K(Zhou et al., 2019) and video recognition on Kinetics-400(Kay et al., 2017), which are also an important metrics for judging our architecture.

ACKNOWLEDGEMENT

This work was partly supported by JSPS KAKENHI Grant Number 20H04165.

REFERENCES

- Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., and Schmid, C. (2021). Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846.
- Beal, J., Kim, E., Tzeng, E., Park, D. H., Zhai, A., and Kislyuk, D. (2020). Toward transformer-based object detection. *arXiv preprint arXiv:2012.09958*.
- Cai, Z. and Vasconcelos, N. (2018). Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162.
- Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer.
- Chen, X., Wu, Y., Wang, Z., Liu, S., and Li, J. (2021). Developing real-time streaming transformer transducer for speech recognition on large-scale dataset. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5904–5908. IEEE.
- Dai, Z., Liu, H., Le, Q. V., and Tan, M. (2021). Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34:3965–3977.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.
- Dong, L., Xu, S., and Xu, B. (2018). Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888. IEEE.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
- Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., and Feichtenhofer, C. (2021). Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835.
- He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Hoffer, E., Ben-Nun, T., Hubara, I., Giladi, N., Hoefler, T., and Soudry, D. (2020). Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8129–8138.
- Hu, H., Zhang, Z., Xie, Z., and Lin, S. (2019). Local relation networks for image recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3463–3472.
- Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al. (2017). The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.
- Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

- Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018). Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.
- Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al. (2022). Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019.
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022.
- Peng, C., Xiao, T., Li, Z., Jiang, Y., Zhang, X., Jia, K., Yu, G., and Sun, J. (2018). Megdet: A large mini-batch object detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6181–6189.
- Polyak, B. T. and Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855.
- Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., and Huang, X. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.
- Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., and Dollár, P. (2020). Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10428–10436.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.
- Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.
- Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Strudel, R., Garcia, R., Laptev, I., and Schmid, C. (2021). Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272.
- Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., et al. (2021). Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14454–14463.
- Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.
- Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.
- Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. (2021). Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578.
- Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500.
- Zhang, P., Dai, X., Yang, J., Xiao, B., Yuan, L., Zhang, L., and Gao, J. (2021). Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3008.
- Zhang, Z., Zhang, X., Peng, C., Xue, X., and Sun, J. (2018). Exfuse: Enhancing feature fusion for semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 269–284.
- Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., and Torralba, A. (2019). Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321.