Semi-supervised Transformer with FPN for Bikes parts detection

Junhao Niu,Xidian University,kaelthas01@163.com,Yu Gu,Xidian University,guyu_07@163.com,
Luyao Nie,Xidian University,15102908628@163.com,Chao You,Xidian University,cy_chaoyou@163.com

Abstract

The purpose of traditional object detection tasks is to detect the position of the target in the picture. But in DelftBikes dataset, which has 10,000 bike photographs in real-world, with 22 densely annotated parts per image, where some parts may be missing, we can also detect the location of missing parts on the bike. This is a very challenging task, because of dense annotations on images and overlapping parts on bikes. For our many experimental results, it shows that object detectors can hallucinate and detect part of missing objects and locate their expected position. We combine Swin Transformer and FPN to construct a object detector, and such detector achieves good results on DelftBikes dataset. Further more, we apply some tricks and pseudo labels obtained from different models to train our detector, and get the best mAP 0.304.

1. Introduction

Deep learning has been widely used in computer vision, and automatically positioning and detecting objects in images is also one of the most important applications of computer vision. However, deep learning object detectors can produce illusions and detect missing objects, and may even accurately locate expected but non-existent locations. This is particularly problematic for applications that rely on visual component verification. Detecting non-existent objects is particularly unfavorable for automatic visual component verification or visual verification applications. The erroneous detection of objects produced by the depth detector may be due to the sensitivity to the absolute position in the image and also affected by the context of the scene.

The data set for this competition is DelftBikes[4], which is a novel and specially created visual object part of the case study to generate illusion verification data set. The data set has a total of 10,000 bicycle images, including 8,000 in the training set and 2,000 in the test set. Each image in the training set is labeled with the bounding box positions of 22 different components.

In response to the above problems, we try a variety of basic target detection models, and typical models will be introduced in recent work. In addition to experimental selection of models, visual observations of training set labels are also performed. Due to the large size difference of 22 types of parts and the large overlap probability, we divided the 22 types of parts into 2 parts for training according to the standard of small size difference and low overlap rate, and added pseudo-labels for iterative training, and finally got the best The results.

2. Related Work

We will introduce some classic models in the field of target detection in this section.

Faster R-CNN[10] improves the region-based CNN baseline. It uses a new RPN[13] (Regional Proposal Network), which is a fully convolutional network that can effectively predict regional proposals with a wide range of scales and aspect ratios. RPN shares full image convolution features and a set of common convolution layers with the detection network, thereby speeding up the generation of regional recommendations. In addition, a new method for targets of different sizes is to use multi-scale anchor points as a reference. This anchor point can greatly simplify the process of generating suggestions for areas of various sizes. The area recommendations are parameterized relative to the reference anchor box. Then measure the distance between the predicted frame and its corresponding ground truth frame to optimize the position of the predicted frame.

RetinaNet[14] is a unified network structure composed of a main network and two sub-networks with designated tasks. The main network is responsible for calculating the convolutional features of the entire input image. The first sub-network further calculates the output of the main network to complete the target classification; the second sub-network is responsible for the bounding box regression. Its main contribution is to propose a loss function: Focal loss[6]. The main purpose is to solve the problem of equilibrium between easy-to-classify samples and difficult-to-classify samples, not just to solve the problem of sample imbalance (in terms of quantity). That is to say, the contribution of easy-to-classify samples to loss is reduced, and the contribution of difficult-to-classify samples to loss is increased.

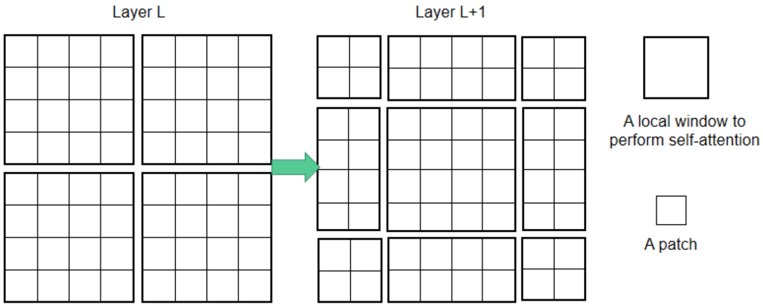

Figure 1. An illustration of the shifted window approach for computing self-attention in the proposed Swin Transformer architecture.

BorderDet[9] is based on the FCOS[12] detection architecture and is a cascaded prediction detector. It mainly adds a BAM module to the prediction head of the feature pyramid. First predict the preliminary bbox prediction and preliminary classification prediction, and then input the preliminary bbox prediction into the BAM module to obtain Border classification prediction and Border bbox prediction. The prediction uses 1x1 convolution as before, and finally the two results are unified for output.

DEtection TRansformer[15] is divided into four parts, namely a CNN backbone, Transformer Encoder, Transformer Decoder, and the final prediction layer FFN. In DETR, a series of object query vectors are responsible for detecting objects at different spatial positions. Each object query interacts with the spatial visual features encoded by the convolutional neural network (CNN), and adaptively collects information from the spatial position through the co-attention mechanism, and then estimates the bounding box position and object category.

The biggest contribution of Swin Transformer[7] is to propose a backbone that can be widely used in all computer vision fields, and most of the hyperparameters commonly found in CNN networks can also be manually adjusted in Swin Transformer, such as the number of network blocks that can be adjusted, each The number of layers of a block, the size of the input image, and so on. The Swin Transformer Block is the core of the algorithm. It consists of a window multi-head self-attention (W-MSA) and a shifted-window multi-head self-attention (SW-MSA) layer. -MSA) composition.

3. Method

3.1. The Swin Transformer Architecture

The main method we used in the competition is the Swin Transformer architecture [7], which has achieved good results on other object detection datasets. An overview of the Swin Transformer architecture is presented in Figure 2. At first, we use a patch splitting module to splits the input RGB image into some small patches, like ViT[2]. Each patch is treated as a “token” and its feature is set as a concatenation of the raw pixel RGB values. Due to limited computing resources, we set patch size as 4×4 and thus the feature dimension of each patch is $4 \times 4 \times 3 = 48$. A layer, serve as linear embedding layer, is applied on this raw-valued feature to project it to an arbitrary dimension C. Several Transformer blocks with modified self-attention module called **Swin Transformer blocks**, are applied on patch tokens above. As the network gets deeper, the number of tokens is reduced by **patch merging layers** to produce a hierarchical representation. The first patch merging layer concatenates the features of each group of 2×2 neighboring patches, and applies a linear layer on the 4C-dimensional concatenated features. This reduces the number of tokens by a multiple of $2 \times 2 = 4$ ($2\times$ downsampling of resolution), and the output dimension is set to 2C.

Swin Transformer blocks are applied afterwards for feature transformation, with the resolution kept at a fixed value. The Swin Transformer defined some stages jointly produce a hierarchical representation, with the same feature map resolutions as those of typical convolutional networks, e.g., VGG[11] and ResNet[3]. Finally, we combine the Swin Transformer architecture extracting features and FPN[5] to locate bicycle parts in images. As a result, the method we used can solve the vision task in this competition well.

3.2. Swin Transformer block

To replace the standard multi-head self attention (MSA) module in a Transformer block, Swin Transformer uses a module based on shifted windows, with other layers kept the same. The basic block consists of a shifted window based MSA module, followed by a 2-layer MLP with GELU nonlinearity in between. A LayerNorm (LN) layer is applied before each MSA module and each MLP, and a residual connection is applied after each module.

The Swin Transformer block computes self-attention within local windows for efficient modeling. The windows are arranged to evenly partition the image in a non-

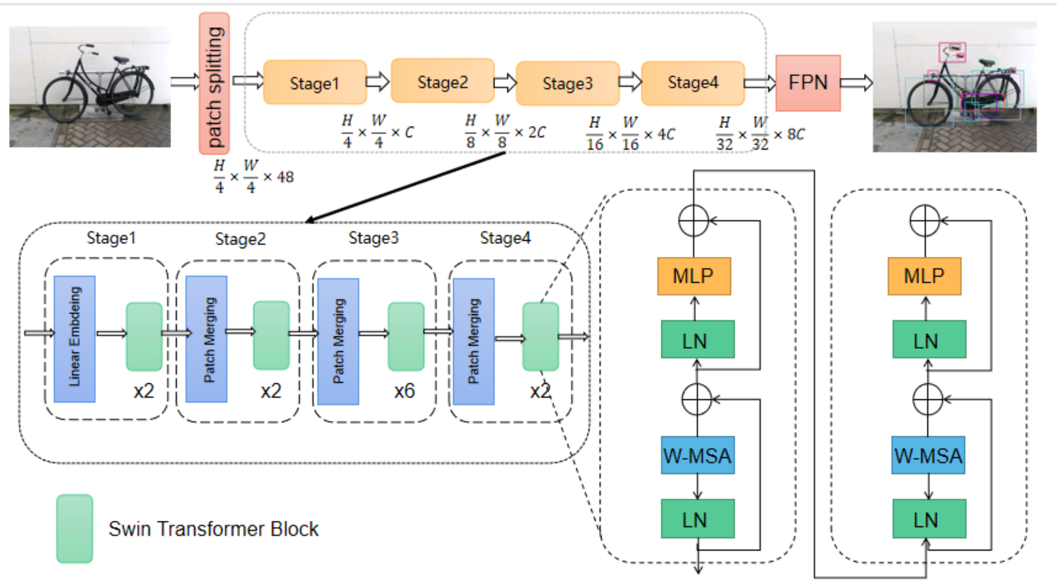
patch splitting
Stage1
Stage2
Stage3
Stage4
FPN
$\frac{H}{4} \times \frac{W}{4} \times 48$
$\frac{H}{4} \times \frac{W}{4} \times C$
$\frac{H}{8} \times \frac{W}{8} \times 2C$
$\frac{H}{16} \times \frac{W}{16} \times 4C$
$\frac{H}{32} \times \frac{W}{32} \times 8C$
Linear Embdeing
Patch Merging
x2
x6
MLP
LN
W-MSA
Swin Transformer Block

Figure 2. The architecture of our object detector with a Swin Transformer (Swin-T) and FPN. Four stages in Swin Transformer are a little different, but two successive Swin Transformer Blocks are the same in each stage.

overlapping manner. Supposing each window contains M×M patches, the computational complexity of a global MSA module and a window based one on an image of h×w patches are:

$$\Omega(\text{MSA}) = 4hwC^2 + 2(hw)^2C, \qquad (1)$$

$$\Omega(\text{W-MSA}) = 4hwC^2 + 2M^2hwC, \qquad (2)$$

where the former is quadratic to patch number hw, and the latter is linear when M is fixed (set to 7 by default). Global self-attention computation is generally unaffordable for a large hw, while the window based self-attention is scalable.

The first module in Figure 1 uses a regular window partitioning strategy which starts from the top-left pixel, and the 8 × 8 feature map is evenly partitioned into2 × 2 windows of size 4 × 4 (M = 4). The second module adopts a windowing configuration that is shifted from that of the preceding layer, by displacing the windows by (M/2, M/2) pixels from the regularly partitioned windows.

W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

3.3. Architecture Select

The Swin Transformer architecture hyper-parameters of these model variants are:

- Swin-T: C=96, layer numbers={2,2,6,2}
- Swin-S: C=96, layer numbers={2,2,18,2}
- Swin-B: C=128, layer numbers={2,2,18,2}
- Swin-L: C=192, layer numbers={2,2,18,2}

where C is the channel number of the hidden layers in the first stage.Swin-T, Swin-S and Swin-L, which are versions of about 0.25×, 0.5× and 2× the model size and computational complexity, respectively. We select **Swin-T** as our baseline and such method reaches the best result in our experiments by using some appropriate strategies.

3.4. Feature Pyramid Networks

FPN(Feature Pyramid Networks)[5] is now used as object detection head in many methods frequently. As a multi-scale object detection method, it can be effectively combined with Swin Transformer to achieve better results on object detection task than other models. Thus, we apply FPN on Swin Transformer to solve the problems in this competiton.

4. Experiments

We first used DETR[15] for training, and set batch size as 16 to train 100 epochs. The training data consists of all 8000 images and their labels given by the official. Later, we found that the effect of the 50 epochs was better than that of the 100 epochs. We later tried to use the Swin Transformer[7] model for training. We choose six different sizes between (800, 1000) and (480, 600) for multi-scale training. We set the batch size as 16 and trained 50 epochs. Its score also exceeds the DETR's score.

The competition needs to detect as many parts as possible, but the part bbox coincides with a higher degree. We

Method	Single model	+Soft-nms	Two parts	Pseudo label	Two parts+Pseudo label
Faster RCNN	0.212	0.232		0.241	
RetinaNet	0.231	0.250	0.251		
Borderdet	0.249	0.252			
DETR	0.252	0.269	0.288	0.292	0.299
Swin Transformer	0.272	0.287	0.293	0.302	0.304

Table 1. Comparison of experimental results on the DelftBikes dataset.

used soft-NMS[1] instead of NMS[8] for prediction and our score improved. By visualizing results, we found that the detection result in some dense areas, such as handlebars and brakes, is offset. We decided to divide the 22 types of labels into two parts to train the prediction results separately. We divide into two parts to make the bboxes of objects with similar features overlap as little as possible. In order to avoid the model from confusing two similar objects, we put objects with similar features but not clustered together in the same part. After dividing into two parts, we still trained the DETR model and the Swin Transformer model respectively. Finally, it was found that the effect of using the two parts to train and test separately will be better than the original 22 types of training together.

Our experimental results are shown in Table 1 , we found that there were more bboxes predicted by the result of DETR, and the result of Swin Transformer was lacking but the accuracy was higher. Then we tried to use the DETR test results and only extracted bboxes with scores higher than 0.5 as pseudo-labels to train Swin Transformer. We found that the Swin Transformer after pseudo-label training works better.

We tried some other models. First, we used Faster RCNN's[10] single model, plus soft-NMS and pseudo-labeled models for training. Then we also used RetinaNet's[14] single model, plus soft-NMS, and the model after training in two parts for testing. In addition, we also used Borderdet's[9] single model and soft-NMS model for training and testing. Finally, we conclude that using Swin Transformer to divide into two parts and add pseudo-labels is the best.

In order to achieve better results, we then used (1)the Swin Transformer testing results to extract bboxes with scores higher than 0.5 as s-pseudo-labels, (2)the s-pseudo-label to train DETR and update the previous pseudo-label, (3) updated pseudo-label to train the Swin Transformer to get the new testing result, and update a new round of s-pseudo-label. Through this continuous repetition, we updated a total of 10 rounds of pseudo-labels and s-pseudo-labels and finally got the best results. All experimental results obtained by methods are shown in Table 1.

References

[1] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In Proceedings of the IEEE international conference on computer vision, pages 5561–5569, 2017.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. 2020.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[4] Osman Semih Kayhan, Bart Vredebregt, and Jan C van Gemert. Hallucination in object detection–a study in visual part verification. arXiv preprint arXiv:2106.02523, 2021.

[5] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2117–2125, 2017.

[6] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, pages 2980–2988, 2017.

[7] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. 2021.

[8] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In 18th International Conference on Pattern Recognition (ICPR'06), volume 3, pages 850–855. IEEE, 2006.

[9] Han Qiu, Yuchen Ma, Zeming Li, Songtao Liu, and Jian Sun. Borderdet: Border feature for dense object detection. In European Conference on Computer Vision, pages 549–564. Springer, 2020.

[10] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28:91–99, 2015.

[11] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. Computer Science, 2014.

[12] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF international conference on computer vision, pages 9627–9636, 2019.

[13] Francesco Zammori and Roberto Gabbrielli. Anp/rpn: A multi criteria evaluation of the risk priority number. Quality

and Reliability Engineering International, 28(1):85–104, 2012.

[14] Hongkai Zhang, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Cascade retinanet: Maintaining consistency for single-stage object detection. arXiv preprint arXiv:1907.06881, 2019.

[15] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159, 2020.