Experiment Report AND Improvements of the Vision Transformer(ViT)

Xiaoyu Ma*
71121117
matthew@seu.edu.cn

Abstract

This report mainly consists of two parts: an experimental report and a course summary.

In the experimental report section, I attempted to use two different models to complete the target task under similar conditions, and then compared the results. I found that under limited resource conditions, the model based on YOLOv8 outperformed the one based on DETR across various metrics.

In the course summary section, I summarized and analyzed the improvement strategies for the ViT model, including the SwinTransformer that can extract features at different scales, and the IDMM method that can train ViT under limited resource conditions. Additionally, to verify the accuracy of SwinTransformer, I trained the SwinTransformer-tiny model and the ResNet18 model on the Garbage dataset, and demonstrated the superiority of SwinTransformer under the same training conditions.

Report of Experiment(A)

1 Experiment Content

1.1 Background

In the process of fabric production, various factors can lead to defects, which need to be detected to ensure quality. However, manual inspection is easily affected by subjective factors and can damage eyesight. Moreover, the complex types of fabric defects and the difficulty of identification have been a technical bottleneck in the industry. This task requires the development of an efficient and reliable machine vision algorithm to improve detection accuracy and efficiency, reduce reliance on manual labor, and not only detect the presence of defects but also provide their location and category.

*Southeast University, Nanjing, China.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

1.2 Dataset

The 9576 images are divided into training/validation/testing sets in the ratio of 7:1:2. The training set has 6703 images. The validation set has 957 images. The testing set has 1916 images.

The dataset contains a total of 20 classes of target objects to be detected.

2 Methods

2.1 YOLOv8

Ultralytics YOLOv8 is a cutting-edge, state-of-the-art (SOTA) model that builds upon the success of previous YOLO versions and introduces new features and improvements to further boost performance and flexibility. YOLOv8 is designed to be fast, accurate, and easy to use, making it an excellent choice for a wide range of object detection and tracking, instance segmentation, image classification and pose estimation tasks.

2.2 DETR

DETR[1]is an innovative end-to-end object detection algorithm proposed by Facebook AI Research and CERN/Université Paris-Saclay in 2020. It abandons many components of traditional object detection and adopts an end-to-end learning approach directly from the input image to the object category and location.

DETR utilizes the transformer encoder-decoder structure, fully leveraging the advantages of transformers in modeling long-range dependencies, which allows it to better capture the contextual relationships between objects. Additionally, DETR does not require predefined anchor boxes, but instead dynamically learns the bounding boxes for each target, eliminating the need for manually designed anchors and improving the model's generalization ability.

3 Experimental Details

3.1 Brief Introduction

We choose YOLOv8 to complete our experiment, since it's easy to use it to finish the experiment. We only did modification on the dataset but not modify the structure because both of us didn't realy be familiar with detection especially the about small things. To ensure the workload for this experiment, we separately trained DETR and YOLOv8 under the same external conditions, using only different pre-trained models. We also tried to ensure that the GPU usage was similar under the same single-training batch size condition, and then compared the results.

3.2 Details

For the YOLOv8 model, we used the pre-trained *yolov8m.pt* model and trained it with images of size 1280×1280, in order to capture more detailed information.

For DETR, due to the larger GPU consumption of the Transformer architecture, we used the rtdetr-l.pt pre-trained model and trained it with images of size 640×640.

All other training parameters were kept the same for both models. Both models were trained for 100 epochs with a batch-size of 64, on 4 $\times$ NVIDIA RTX3090 GPUs.

4 Experimental Results and Analysis

Here is the comparison between DETR based model and YOLO based model the results are showed in table 1. We can found that Yolo based model get better result on all evaluating indicators. It may because the limited size of images, but unfortunately we cannot expand the size since the memory limitation of GPUs. But we confirm that the two models are similar in terms of GPU consumption.

But we can know that, when the memory of GPU is limited, YOLOv8 can perform better than DETR.

Model	Pretrain	imgsz	precision	recall	mAP50	mAP95
DETR	rtdetr-l	640	0.46033	0.39831	0.3728	0.19275
YOLOv8	yolov8m	1280	0.58288	0.45122	0.50153	0.2405

Table 1: The comparision of DETR based model and YOLOv8 based model

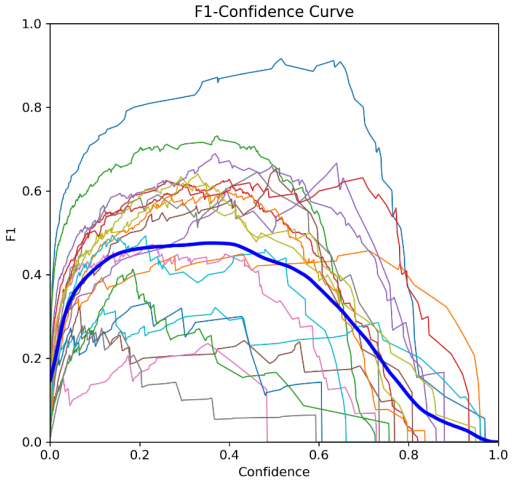

Figure 1: F1-confidence curve of YOLOv8 based model

Last but not least, we get the image of F1-confidence curve when validate YOLOv8 based model(see Figure 1).

The different colored curves represent different classes, and the thickest line represents the average case. This figure is not for the purpose of analyzing the curves of specific classes, so the class labels have been removed. The aim is to focus on the fact that there are significant differences between the curves of different classes, and their performance in terms of F1 accuracy also varies greatly.

The significant differences between the curves of different classes may be due to the inherent differences in the detection difficulty of these classes. However, we also noticed that the number of training samples for different class targets is unbalanced. This suggests that there is room for further data augmentation or model improvement to enhance the overall accuracy.

The Second Part

1 Background

The Vision Transformer[2] also known as ViT, is an image classification model based on the Transformer[3] architecture. It has gained widespread attention and research in the field of computer vision. Unlike traditional Convolutional Neural Networks, ViT treats an image as a sequence or a set of vectors and utilizes the self-attention mechanism of Transformers to learn global relationships and feature representations within the image.

The core idea of ViT is to divide the input image into a series of image patches and convert each patch into a vector representation. These vectors are then passed as input sequences to the Transformer encoder for processing. In the Transformer encoder, the self-attention mechanism is used to learn the interdependencies between the image patches while performing feature extraction and representation learning. Finally, after several layers of Transformer encoders, ViT outputs a global representation of the image, which can be used for tasks such as image classification.

Despite achieving remarkable results in image classification tasks, ViT also faces certain challenges and limitations. Firstly, since ViT processes images based on fixed-sized image patches, it may not adapt well to pixel-level tasks such as image segmentation or object detection. Secondly, the computational complexity of ViT's attention module is proportional to the square of the number of image patches. Therefore, having a large number of patches can lead to excessive computational requirements, while having too few patches can affect the capture of detailed features. Finally, ViT has a high demand for training data and typically requires large-scale datasets for pretraining, such as JFT-300M or at least ImageNet. These datasets require significant computational resources and time for collection and annotation, limiting the feasibility for ordinary researchers or projects.

2 Related Works

Swin Transformer[4] is a visual model based on the Transformer architecture, designed to address the challenges of scale invariance and image patch size selection encountered by ViT. By introducing hierarchical representations and window-based attention mechanism, Swin Transformer offers an effective approach to handle images at different scales and overcomes the limitations of fixed-sized image patches in ViT. The hierarchical representations enable Swin Transformer to capture features at different levels, thereby enhancing the model's adaptability to scale variations. The window-based attention mechanism replaces fixed-sized image patches, allowing Swin Transformer to flexibly adjust the window size to accommodate objects of different scales.

To address the issue of high training and data costs in ViT, IDMM[5] conducted research on a parameter instance discrimination-based method to train ViT on a limited amount of data(e.g., 2040 images), and provided theoretical analysis. Additionally, in terms of transfer learning, representations learned using this method from small datasets can even improve upon large-scale ImageNet training.

3 Methods

This section describes the model architecture of Swin Transformer and emphasizes its comparison with ViT, highlighting the optimized designs in terms of multi-scale processing and reducing computational complexity.

3.1 Basic Architeure

The architecture of Swin Transformer (see Figure 2) is an innovative development based on the Transformer model, incorporating key designs such as hierarchical representations, window-based attention mechanism, mixed-precision training, and layer interconnections. Through hierarchical representations, Swin Transformer can capture image information at different scales, thereby enhancing its adaptability to scale variations. The window-based attention mechanism overcomes the limitation of fixed-sized image patches in ViT, enabling the model to flexibly handle objects of different scales.

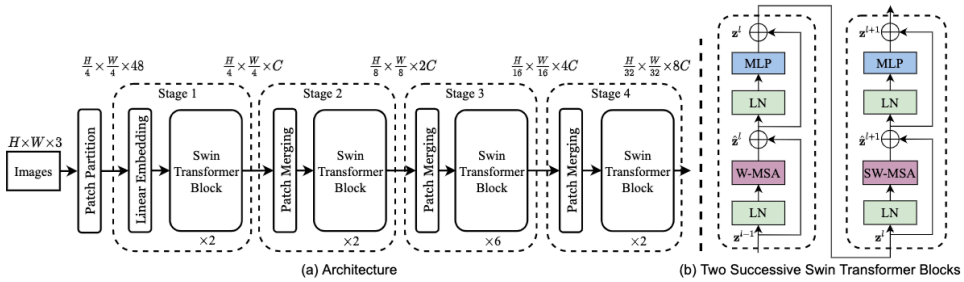

Figure 2: (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks

3.2 W-MSA and Patch Merging

Compared to the attention mechanism used in ViT, Swin Transformer employs window based attention(W-MSA), which significantly reduces the computational complexity of the attention mechanism. Supposing each window contains $M \times M$ patches, the computational complexity of a global MSA module and a window based one on an image of $h \times w$ patches are

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

$$\Omega(W-MSA) = 4hwC^2 + 2M^2hwC$$

Due to space limitations, a detailed description of the computation process is not provided here. You can click and refer to my Blog for a specific explanation of the computation process.

After W-MSA, to generate multiple levels of resolutions as the network deepens, Swin Transformer utilizes the Patch Merging layer. In the first Patch Merging stage, for an input size of $(C, W/4, H/4)$, where C represents the number of channels and W, H represent the width and height respectively, 2×2 patches are grouped together. The channel dimension is increased to $4C$, and the resolution is reduced to $1/4$, resulting in an output size of $(4C, W/8, H/8)$, similar to a feature reshape operation. Subsequently, the dimension is further reduced to $2C$, resulting in an output size of $(2C, W/8, H/8)$. In stage 3 and stage 4, the output resolutions are $(4C, W/16, H/16)$ and $(8C, W/32, H/32)$ respectively.

3.3 Shifted Window based Self-Attention

The fixed window approach lacks inter-window connectivity, which limits the expressive power of the model. To address this, the shifted window partition method is introduced. In this method, layer l uses W-MSA, while layer $l + 1$ uses shifted window partitioning (SW-MSA), which generates windows that overlap with the boundaries of the windows in layer l(see in Figure 3).

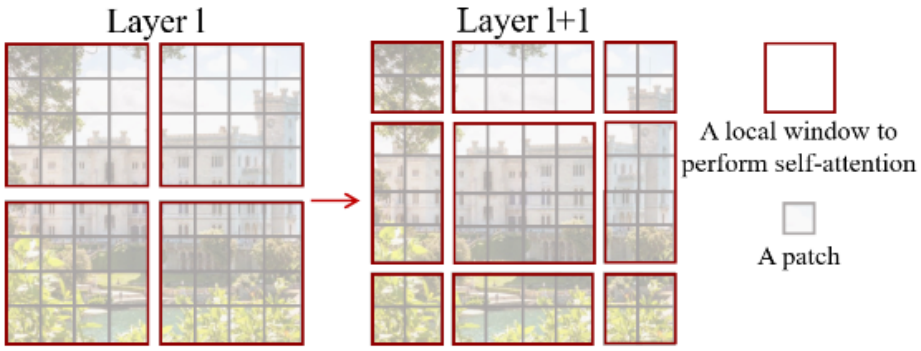

Figure 3: An illustration of the shifted window approach for computing self-attention in the proposed Swin Transformer architecture.

By using shifted window partitioning, more windows are generated, including smaller individual windows. To ensure the same number of windows as the conventional window partitioning, the authors design and employ cyclic shifts. However, after the shift, some regions that are not adjacent in the feature map end up in the same window. Attention calculation should not be performed between these regions, so a mask is introduced to address this issue.

4 Experimental Details

4.1 Basic Task

To ensure the classification capability of Swin Transformer, I did the experimental on Dataset *Garbage265*. It's a collection of 147,674 life garbage images with Chinese labels.

I compared the capability of Resnet18[6] and Swin Transformer. Swin Transformer is initialized from microsoft/swin-tiny-patch4-window7-224 on huggingface, which is trained on ImageNet-1k at resolution 224×224. The model of ResNet is trained on the same condition. Both of them are fine-turned on *Garbage265*.

4.2 Details

Both of the models are fine-turned with the same hyperparameters. I trained them on a Nvidia GeForce RTX 3090(24G) for 20 epochs with bz=64. I used the Adam optimizer and learning rate is set to 1e-3. CrossEntropy is selected to be the loss function.

For the two pretrained models, I only changed the last Linear layer to ensure the output of models have the size $(B, 265)$. While the optimizer is serviced on all parameters of the model.

Last but not least, RandomRotation, Centercrop and Normalize are used to achieve data enhancement.

5 Experimental Results and Analysis

I only compute the Top-1 accuracy of my fine-turned models. The Swin Transformer get the top-1 accuracy of **92.3%** and the ResNet18 get the top-1 accuracy of **89.6%** on test dataset.

It seems that Swin Transformer is better on accuracy, while Resnet18 is more small in Parameters(M), see table 2 for more detalis.

Model	Top-1(%)	Parameters(M)
ResNet18	89.6	11.7
Swin-tiny	92.3	27.6

Table 2: The comparision of ResNet18 and Swin-tiny results

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *CoRR*, abs/2005.12872, 2020.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv: Computer Vision and Pattern Recognition,arXiv: Computer Vision and Pattern Recognition*, Oct 2020.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, AidanN. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Neural Information Processing Systems,Neural Information Processing Systems*, Jun 2017.

[4] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2021.

[5] Yun-Hao Cao, Hao Yu, and Jianxin Wu. *Training Vision Transformers with Only 2040 Images*, page 220237. Jan 2022.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.