Classification of Fine-Art Paintings Genre Using Deep Neural Network

Tianjian Ni (tn2151)   Dowon Yang (dy2242)

Abstract

In the project, we use the power of deep neural networks to classify famous artworks' genres. We experiment with different computer vision models such as Vision Transformer (ViT), Inception Transformer (iFormer), and ResNet-50. The results show that ResNet-50 works the best in successfully classifying the genres among the tested models.
GitHub: https://github.com/TianjianNi/ViT-on-WikiArt

Introduction

Fine art paintings are usually displayed in different rooms based on their genres when visiting art galleries. These genres are classified by art historians. But can we teach a computer to do the same task? Deep neural networks have been heavily used to extract features and to conduct the classification of objects in computer vision tasks and have achieved good results. In this project, we hope to transit the deep learning models into a new area which is to classify styles of fine-art paintings.

The dataset used is called "Best Artworks of All Time". The dataset is scraped from artchallenge.ru and collects the artworks from 50 influential artists of all time. For this project, we select 10 artists whose works are the best representatives of 10 genres. The list of artists and genres is in Figure 1. We will use Vision Transformer (ViT), Inception Transformer (iFormer), and ResNet-50 to classify the artworks.

Name	Genre	Paintings
Pablo Picasso	Cubism	439
Rembrandt	Baroque	262
Alfred Sisley	Impressionism	259
Titian	High Renaissance	255
Amedeo Modigliani	Expressionism	193
Andy Warhol	Pop Art	181
Frida Kahlo	Primitivism; Surrealism	120
Andrei Rublev	Byzantine Art	99
Piet Mondrian	Neoplasticism	84
Jackson Pollock	Abstract Expressionism	24

Figure 1: Artist Names and Genres.

Literature Survey

ResNet. (He et. al. 2015) introduced the ResNet-50, a groundbreaking architecture. Traditionally when the deep neural networks become deeper, the problem of vanishing/exploding gradients will occur. ResNet-50 network addresses the problem by introducing residual connection, where shortcut connections between the input and the output are established. Figure 2 provides an example of a residual connection. The ResNet-50 network is used in different tasks such as Image Classification, Object Detection and Localization, and Transfer Learning.

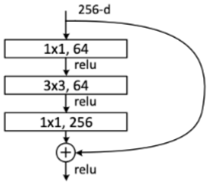

Figure 2: a building block for ResNet-50

Vision Transformer (ViT). The transformer architecture (Vaswani et. al. 2017) introduced the concept of attention mechanism, allowing to model of long-range dependencies. After the transformer was actively used in the NLP area for a few years, (Dosovitskiy et. al. 2021) introduced the Vision Transformer, which is inspired by the transformer architecture. The Vision Transformer uses a transformer encoder to replace convolutional blocks and apply patches along with position embeddings to feed into the encoder. Figure 3 provides an example of a Vision Transformer architecture and a transformer encoder. Nowadays Vision Transformers are applied in areas such as Image Classification, Semantic Segmentation, and Instance Segmentation.

Inception Transformer iFormer. (Si et. al. 2022) introduced the Inception Transformer (iFormer). It is a novel vision transformer that is a fusion of both convolutional layers and attention layers. The concept of the Inception mixer can graft the advantages of Transformers for capturing low-frequency information and the advantages of CNN for capturing high-frequency information. The Inception Transformer is applied to different tasks such as Image Classification, Object Detection, and Segmentation.

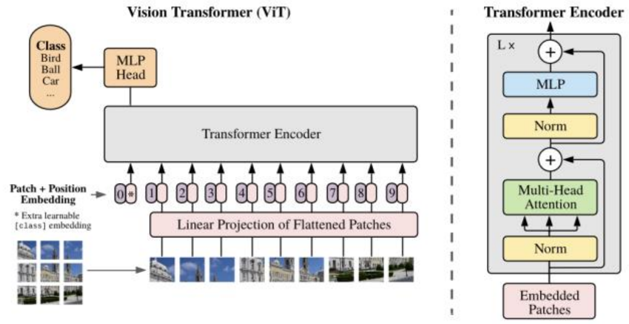

Figure 3: Vision Transformer Architecture and Transformer Encoder

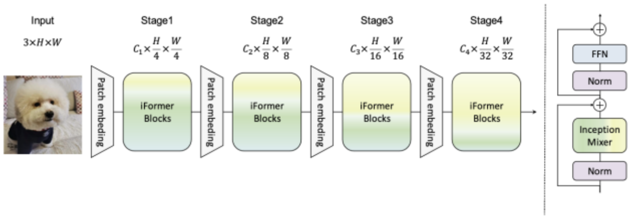
Input
3×H×W
Stage1
$C_1\times\frac{H}{4}\times\frac{W}{4}$
Stage2
$C_2\times\frac{H}{8}\times\frac{W}{8}$
Stage3
$C_3\times\frac{H}{16}\times\frac{W}{16}$
Stage4
$C_4\times\frac{H}{32}\times\frac{W}{32}$
Patch embedding
iFormer Blocks
FFN
Norm
Inception Mixer
Norm

Figure 4: The overall architecture of Inception Transformer and details of the iFormer block

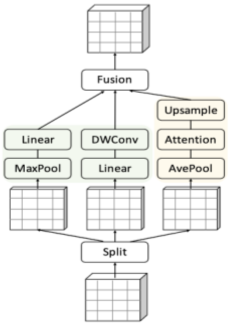

Figure 5: The details of Inception mixer

Technical Details

High-performance computing HPC. For the experiments in the paper, we run all the computations on Greene, NYU's primary high-performance computing (HPC) cluster. We also implement a data parallelism approach to train the deep learning models. When using data parallelism, we replicate the model on each GPU, send portions of the data batch to the GPUs, compute in parallel in each GPU, and gather the result at the end. *DistributedDataParallel* is the data parallelism method we select to use. Each GPU runs a process and uses the AllReduce operation to compute gradient summation across all GPUs. Figure 6 provides an example of how AllReduce works in *DistributedDataParallel*. One advantage of *DistributedDataParallel* compared with other data parallelism methods is that GPUs do not have to be on the same node, they can belong to different nodes connected through a communication infrastructure. Due to the large volumes of HPC access requests near the end of the semester, we use 2 GPUs to run *DistributedDataParallel* in this project to avoid days of waiting for HPC resources.

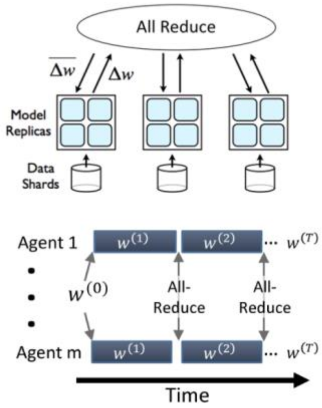

Figure 6: AllReduce in *DistributedDataParallel*

Loss Function. The cross-entropy loss is applied to compare the predicted probability distribution with the true distribution (one-hot encoded ground truth labels). This loss function is well-suited for classification tasks and encourages the model to assign high probabilities to the correct classes.

Optimizer. The AdamW optimizer is used in the project. The AdamW is a common optimizer used by models to classify the ImageNet-1K dataset. The initial **learning rate** is set as 0.0008 with **weight decay** as 0.01. **betas** are the coefficients used for computing running averages of gradient and its square and we set them as (0.9, 0.999).

Num of Workers. To facilitate data loading processes, there are 4 workers to load the images to GPUs. Thus, 4 CPUs will are requested to run the processes.

Epochs. Each model will run 50 epochs to ensure they are warmed up and parameters can be optimized to achieve the best result.

Batch Size. we use a small batch size of 32. *DistributedDataParallel* will load the batch size of 32 in parallel to each of the GPUs and use AllReduce to sum the gradients across all GPUs during backpropagation.

Train Test Split. 85% of the images in the dataset are in the training group and 15% of the images in the dataset are in the validation group.

Seed. To enhance reproducibility, a PyTorch seed will be set.

Experiment Details. We first code a plain vanilla Vision Transformer (ViT) from scratch based on the original paper (Dosovitskiy et al. 2021) and achieve around 33% accuracy in our test run of 10 epochs. We think there were some implementation mistakes. So, we run again using the Vision Transformer model from the Timm library. Both our plain vanilla implementation and the model implementation use 16 as the patch size, 12 as the number of attention heads, and 12 as the number of blocks/depths. Both models achieve a similar result. Then we are assured that our plain vanilla implementation is correct by comparing the results. We also test other deep learning models including ResNet-50 and Inception Transformer (iFormer). The experiments' results will be explained in the next section.

Results

	ViT from Scratch	**ViT**	**ResNet**	**iFormer**
Epoch 10	38.94	27.64	61.54	66.70
Epoch 20	49.26	44.96	74.20	87.10
Epoch 30	49.50	50.61	82.55	79.48
Epoch 40	53.56	49.38	91.76	89.43
Epoch 50	55.15	56.38	95.82	79.60

Figure 7: Output for Training Accuracy (%)

	ViT from Scratch	**ViT**	**ResNet**	**iFormer**
Epoch 10	33.33	29.16	54.86	46.52
Epoch 20	43.05	34.02	58.33	20.83
Epoch 30	41.66	37.50	63.19	43.05
Epoch 40	41.66	41.66	63.19	32.63
Epoch 50	47.91	38.88	72.91	72.22

Figure 8: Output for Validation Accuracy (%)

	ViT from Scratch	**ViT**	**ResNet**	**iFormer**
Epoch 10	24.20	40.15	18.74	45.87
Epoch 20	24.38	39.93	18.89	46.07
Epoch 30	24.12	39.86	18.78	45.95
Epoch 40	24.22	39.93	18.92	45.98
Epoch 50	24.27	40.11	18.85	45.93

Figure 9: Output for Time Running the Epoch (sec)

Since there are 10 classes to classify, a random guess would yield around 10% accuracy. Based on our experiments' results in Figures 7 to 9, we found out that the ResNet-50 produces robust performance in both training and validation accuracies while requiring the smallest time to train. Both training and validation accuracy keep increasing in the training phase and the validation accuracy reaches 70%, which is the highest among all. The Inception Transformer is the slowest to train and has a bottleneck in training accuracy at around 80% while its validation accuracy fluctuates a lot. Both Vision Transformer models have poor performance since their training and validation accuracies cannot pass 50%. But surprisingly our plain vanilla Vision Transformer (ViT) model takes roughly half of the time than the library implementation.

Conclusion

In the original project proposal, we intended to use the WikiArt dataset. There are 2 download links for the dataset. One link has broken access, and another has unlabeled data, which is used to conduct unsupervised learning tasks, especially for generating images using GAN. Then we find the "Best Artworks of All Time" dataset and decide to go with it. We also planned to calculate the Top-3 accuracy since the WikiArt dataset was supposed to give us 27 classes for classification. Since eventually we only classify 10 classes, Top-1 accuracy alone should be enough to show performance. Other than the changes in the image dataset and the performance metrics selection, all the goals from the project proposal were achieved. Through the project, we can fully understand the blocks of the Vision Transformer (ViT) architecture by building a plain vanilla model from scratch. We are also able to experiment with other popular computer vision models and compare their performances.

The results show that the traditional CNN method ResNet-50 can still be robust in some classification tasks and outperform the trended Vision Transformer models.

References

He, K.; Zhang, X.; Ren, S.; Sun, J. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385.

Si, C.; Yu, W.; Zhou, P.; Zhou, Y.; Wang, X.; Yan, S. 2022. Inception Transformer. arXiv: 2205.12956.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; Houlsby, N. 2010. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv: 2020.11929.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. 2017. Attention Is All You Need. arXiv: 1706.03762.

Kaggle. 2019. Best Artworks of All Time. www.kaggle.com/datasets/ikarus777/best-artworks-of-all-time. Accessed: 2023-12-17.

PyTorch. DistributedDataParallel. pytorch.org/docs/stable/generated/torch.nn.parallel.DistributedDataParallel.html. Accessed: 2023-12-17.