An Intermediate Summary for Current Project

Qitao Zhao on 16 March, 2022

This article aims at intermediately summarizing my ongoing work on Gait Recognition.

I started to work on Gait Recognition with *Prof. Xianye Ben* at the beginning of the last semester (Sept. 2021). I was a freshman on research and it was my first project related to deep learning. Before that, I studied Stanford CS231N as an introduction to deep learning, from which I learned deep learning basics, how to use Numy, Pytorch, and I read a series of classic papers (e.g. ResNet, BatchNorm, Transformer).

Equipped with those fundamental knowledge and skills, I began with reading top-tier conference (as well as journal) papers about Gait Recognition during my free time (class schedule was still heavy). I did not participate in a specific project then and it was more like a self-exploring. Till the end of the last semester (Dec. 2021), I had read about 20 papers about Gait Recognition. It may seem weird that for a whole semester, I just read papers but did not do any hands-on work (e.g., conducting experiments, developing models) on Gait Recognition due to a lack of computational resources (I had requested for those though). However, a large amount of paper reading brought me a good understanding of this field as well as deep learning. (I also read papers about Style Transfer, Object Detection and e.t.c., that were needed for course projects.)

Since I observed that, almost all recent state-of-the-art methods manifest a similar design philosophy, they typically use both local (frame-level) feature extraction and global (sequence-level) feature extraction, I thought I could come up with a novel approach to achieve effective temporal (i.e., sequence-level or dynamic) feature aggregation. Considering the virtue of vision transformer (**vit**) to build long-range dependencies, a simple idea came out: I could use vision transformer as global feature extractor. Luckily, no one had done this on Gait Recognition so far. It was a vague idea then.

The biggest challenge to apply vit in Gait Recognition is that vit needs heavy pre-training or large training dataset while Gait Recognition dataset is relatively small. To know more about the recent progress of vision transformer, I focused on it during the winter holiday (Jan - Feb, 2022), reading papers especially about vit trained on small dataset, vit combined with convolutions and e.t.c.. I thought the problem of lacking inductive bias could be alleviated by coupling vit with convolutions. This confomed to using CNNs to extract frame-level at first.

The new semester came (late Feb, 2022). As with still no access to available GPUs, I rent 2 RTX-3090s (then 4 for more capacity) on my own. Only then was I able to run expriments myself! Based on OpenGait I ran several baselines, and gradually I got familiar with this framework. On Feb 23, I developed my first model, named **GaitFormer**.

The idea was quite straightforward. In the first stage, a CNN block is applied to each frame within an input sequence. I assumed early convolutional layers are beneficial: **1.** They bring inductive bias for transformer block, which may ease the training difficulty. **2.** The resolution of each frame is reduced, and therefore later matrix multiplicaton becomes less expensive. In the second stage, high-level features of each frame are flattened as input into vision transformer, which serves as a global feature aggregator. Finally, we use triplet loss and softmax loss as

training signal. The overall structure of GaitFormer is shown in Fig 1. I conducted a series of experiments on **CASIA-B**, which contains 124 subjects (labeled in 001-124), 3 walking conditions and 11 views (0◦, 18◦, ..., 180◦), a relatively small dataset. Empirically, the model easily overfitted the training data. Plus, the overfitting was not sensitive to the model size. The model of various sizes achieved a similar performance while it was far behind the training accuracy. I thought the overfitting might be attributed to the intrinsic property of transformer as described in [1, 2, 3]. Though these papers do not analyze exactly the reason underlying the overfitting, personally speaking, I thought that dense connection operations within transformer blocks might be the cause.

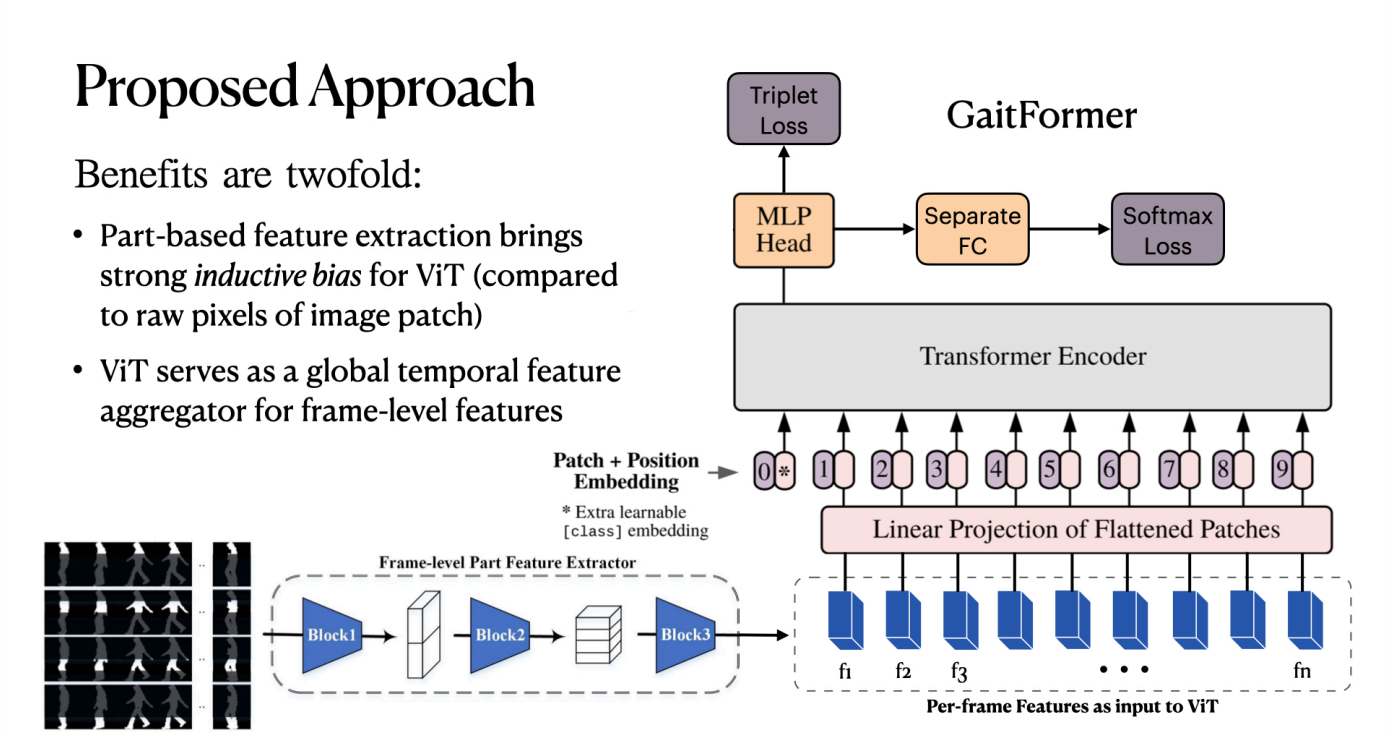

Fig. 1 Structure of GaitFormer.

To alleviate the problem of overfiting, I introduced pooling layer in transformer as [4], which removes redundant as well as expensive self-attention mechanism in transformer block. Besides, I adopted a multi-scale structure. Intuitively, progressively aggregating global information might outperform aggressively doing this only on high-level features. The structure of modified GaitFormer is presented in Fig 2 and the Loss Block is in Fig 3.

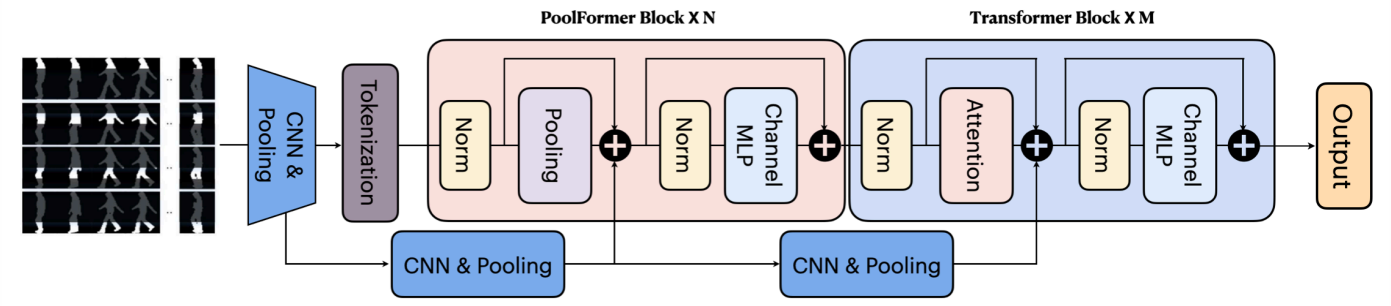

Fig. 2 Structure of GaitFormerV2.

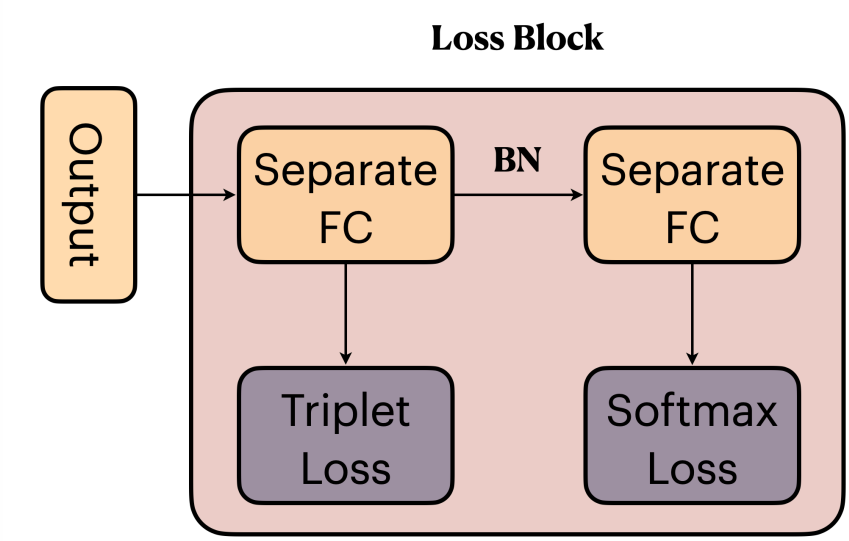

Fig. 3 Loss Block of GaitFormerV2.

To further fuse global and local features, a ResNet-like shutcut was exploited. In first N stages, we use PoolFormer Block instead of Transformer Block to reduce expensive dense-connection opeartions while aggregate information within neiboring frames. In later stages, Transformer Blocks gather global information.

I am now conducting experiments on **GaitFormerV2** and at the same time, applying this in The 3rd International Competition on Human Identificaiton at a Distance 2022.

References

[1] Haotian Yan, Zhe Li, Weijian Li, Changhu Wang, Ming Wu, and Chuang Zhang. Contnet: Why not use convolution and transformer at the same time? CoRR, abs/2104.13497, 2021.

[2] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers and distillation through attention. arXiv preprint arXiv:2012.12877, 2020.

[3] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, Lucas Beyer. How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers. arXiv preprint arXiv:2106.10270, 2021.

[4] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, Shuicheng Yan. MetaFormer is Actually What You Need for Vision. arXiv preprint arXiv:2111.11418, 2021.