CCT (Compact Convolutional Transfomer)

Main Subject: CCT의 간략한 구조 파악하기

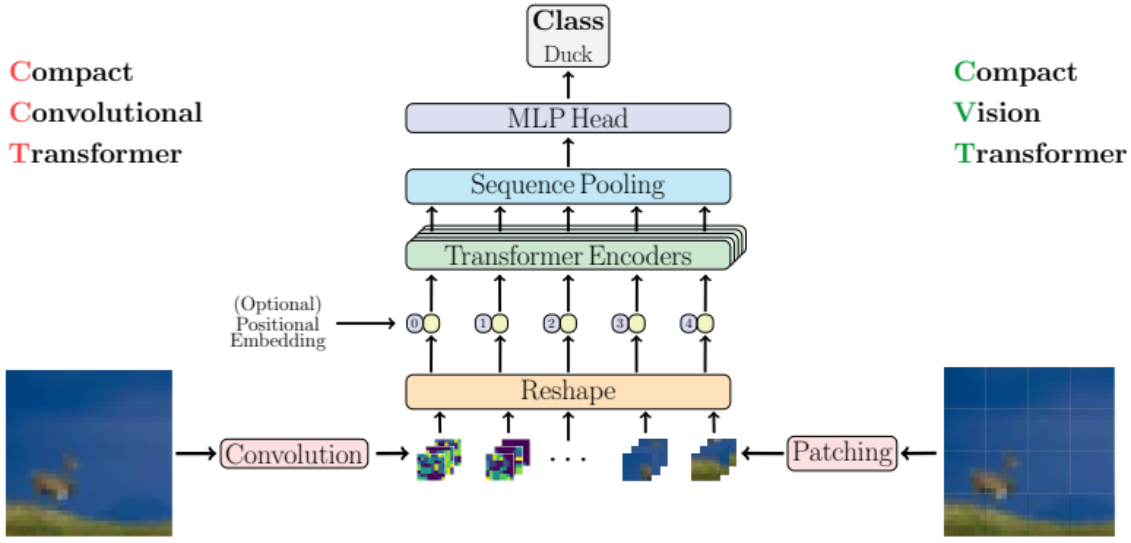

Figure 1: Overview of **CVT** (right), the basic compact transformer, and **CCT** (left), the convolutional variant of

ViT-Lite → ViT의 작고 Compac한 버전, 작은 Resolution을 갖는 CIFAR10에 대해서 90% 이상의 정확도를 기록함.

- Main Contributions

1. Transfomer 기반의 아키텍처를 작은 데이터셋에서(CIFAR10 등)도 좋은 성능을 낼 수 있도록 함.
2. Compact Vision Transformer(CVT) → New sequence pooling strategy (**SeqPool)**제안.
3. Compact Convolutional Transformer(CCT) → Positinal Embedding에 대한 의존성을 낮춤.

Method

CVT는 SeqPool을 통해 class token을 대체하는 것으로 구성됨

CCT는 Patch 기반으로 토큰을 만드는 것이 아닌 Convolutional Layer를 통해 토큰을 생성함. 따라서 Conv Layer를 통해 생성된 토큰은 Projection을 통해 임베딩 되는 것이 아닌 **Pooling**을 통해 가공됨. 또한 원래 ViT에 비해 지역적 정보를 보존하는(패치 간의 관계) 것으로 나타남.

따라서 CVT,CCT에서는 둘 다 Class Token이 없으나, Patch 기반 토큰을 생성하는 CVT는 Positinal Embedding Token이 필요하지만, CCT에서 필요 없음.

- Same backbone, But make small and compact Model

기본적으로 쓰이는 Backbone 구조는 기존의 ViT와 같다 (MHSA, MLA의 Encoder 블럭, Layer Norm, GELU 등..)

ViT-Base 모델을 예로 들어, 12개의 Encoder, 12개의 Attetnion head, D=2048임. 85M parameters, (16x16 patch 기준)

하지만 작고 Compact한 모델을 위해서 본 논문이 제안하는 바는, 2개의 Encoder, 2개의 Attention head, D=128임. (총 0.22M parameters)

- SeqPool

ViT는 Sequential한 결과(Output sequence of tokens)를 하나의 클래스로 판정하기 위해서, 클래스 토큰을 forwarding하거나 Query token을 Classfier에 주는 구조임. 그래서 이 토큰들을 Global Average Pooling하는 경우가 흔했다.

하지만 SeqPool은 **Attention을 기반으로 Output sequence of tokens를 Pooling 하는 방법**이다. 이 아웃풋 시퀀스는 input 이미지에 대한 다양한 정보를 갖고 있으므로, 이 시퀀스가 정보를 잘 보존할 수 있게끔 Pooling하는 방법을 찾으면 성능을 개선할 수 있을 것이라 생각.

Global Average Pooling과 같이 정적인 방법이 아니라 Learnable하게 구성되어 있음

→ Encoder에서 생성된 Sequential Embedding과 Input Data간의 연관성을 찾을 수 있게 해줌, 따라서 SeqPool로 Class token을 대체할 수 있게 됨.

- Convolutioal Tokenizer IN CCT

모델에 inductive bias를 추가하기 위해, patch, embedding을 convolutional block으로 대체함, 간단한 ReLU, MaxPool 구조를 따름.

이를 통해 locally spatitial한 정보를 잘 습득할 수 있게 되고, 이를 통해서 이미지 해상도에 대한 제한적인 요소가 많이 해소된다.

→ Feature map을 토큰으로 쓰겠다는 얘기인가?

Convolution은 Transfomer에게 더 많은 정보의 Token으로 입력될 수 있음.

단, Self-Attention 연산은 입력된 토큰에 따라 많은 시간과 공간 리소스를 요구하므로, Downsampling(필터 사이즈, stride, padding 크기 등..)의 정도가 다른 결과를 만들고, 더 적은 feature map이 computation을 줄여줄 수 있을 것임.

→ Feature map도 가능한 작게 만들어서 토큰의 개수도 좀 줄여서 계산양을 줄이겠다는 소리인 듯..

Conv layer를 통해 지역적인 정보도 쉽게 획득이 가능하므로, Positional Embedding도 필요가 없음, 없앤 채로 실험해 본 결과 좋은 성능을 유지하는 것을 확인함.

CCT Model의 Notation: CCT-7/3x2 = 7개의 Encoder,3x3 필터 사이즈를 가진 2개의 Conv Tokenizer