Shifted Window 분할 기법을 통한 Hierarchical Attention 기반 Vision Transformer 모델의 정확도 개선

서수희○, 신동군
성균관대학교 전자전기컴퓨터공학과
skyb41288@skku.edu, dongkun@skku.edu

Accuracy Improvement of a Hierarchical Attention-based Vision Transformer Model through Shifted Window Partitioning

Soohee Seo○, Dongkun shin
Department of Electrical and Computer Engineering, Sungkyunkwan University

요 약

최근 ViT(Vision transformer) 모델은 컴퓨터 비전 분야에서 이미지 분류, 객체 탐지 등 뛰어난 성능을 발휘하며 주목받고 있다. 하지만 고해상도 이미지 처리에서는 self-attention 연산에서의 입력 길이가 길어지면서, 이차 복잡도 연산으로 계산 비용이 증가한다. 이에 따라, Swin transformer 논문은 window 기반 attention을 도입하여 연산량을 선형적으로 줄이고, window shift를 통해 window간 정보를 공유한다. Swin transformer와 달리, FasterViT는 각 window의 정보를 요약하는 캐리어 토큰을 사용하여 window간 정보 교환을 했으나, 고정된 window 분할 방식을 사용하고 있어서 장거리 의존성을 완벽하게 포착하지 못한다. 본 논문에서는 FasterViT 모델 기반으로 Swin transformer의 shifted window 기법을 활용한 SWHAT(Shifted window hierarchical attention)기법으로 window 간 정보 공유를 강화한다. Imagenet-1k 데이터 세트를 통한 이미지 분류 실험결과, 정확도 81.694%을 달성했으며, 기존 Fastervit 대비 정확도 0.272% 향상을 보였다.

1. 서 론

트랜스포머(Transformer)[1]는 원래 자연어 처리 작업을 하기 위해 도입된 모델이지만, 최근에는 컴퓨터 비전 분야(Computer vision)에서도 뛰어난 성능을 발휘하며 주목받고 있다. VIT[2](vision transformer)는 이미지를 겹치지 않는 패치로 분할한 다음 이 패치들을 토큰으로 처리한다. 이후, 트랜스포머 layer를 반복적으로 수행하며 이미지 분류를 위한 전역 관계를 모델링한다. 이 기술은 CNN과 다르게, 트랜스포머 구조상 토큰들 간의 관계를 잘 추출하여 지역 및 전역 정보를 학습할 수 있다. 이후 여러 ViT 구조 기반 모델들[3]이 등장하여 연구가 활발히 진행되었다.

하지만 ViT는 고해상도 이미지를 입력으로 사용하는 비전 작업에 적용하기 어렵다. 왜냐하면 고해상도 이미지를 입력으로 사용할 때 계산 비용이 매우 크기 때문이다. Self-attention 연산은 입력 길이에 대해 이차적 복잡도를 가지고 있다.

* 이 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.IITP-2017-0-00914, 지능형 IoT 장치용 소프트웨어 프레임워크)

이러한 문제를 해결하기 위해 Swin transformer[4]는 작은 크기의 패치에서 시작하여 더 깊은 트랜스포머 layer에서 인접한 패치를 합치면서 계층적 표현을 구축하는 다중 스케일 구조를 제안하였다. Window 기반 self-attention을 제안하고, shifted window를 통해 다른 영역 간의 상호 작용을 허용하였다.

그림 1과 같이, 각 window 내의 패치 수는 고정되어 있으므로 복잡성은 이미지 크기에 비례하여 선형적으로 증가한다. 이로써, Swin transformer는 이전의 트랜스포머 기반 구조와 대비되며, 이차 복잡성을 해결하고, 다양한 시각적 작업에 적합한 백본으로 사용될 수 있다. 그러나 고정된 window shift 방식으로, 서로 공유된 적이 없는 토큰끼리는 정보 교환이 계속 일어나지 않는다. 한편, FasterViT[5]는 각 트랜스포머 블록에서 짧은 범위와 긴 범위의 공간 의존성을 모두 포착하고 window 간 상호작용을 효율적으로 모델링한다. 그림 1과 같이, 각 window의 요약으로 캐리어 토큰(carrier token)을 학습하고, window 간의 정보를 교환한다. 그러나 FasterViT의 캐리어 토큰이 각

window를 요약하여 모든 window 간의 정보 교환이 일어나지만, 고정된 window 분할 방식을 채택하고 있다. 이 점은 여전히 장거리 의존성을 완벽하게 포착하지 못할 수 있다.

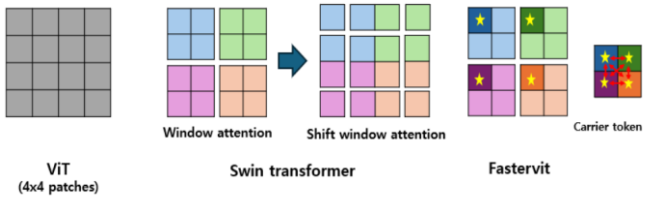

그림 1. ViT 모델 별 패치 window 분할 방식

본 논문에서는 FasterViT의 HAT(Hierarchical attention) layer를 기반으로 Swin transformer의 shifted window 방식을 활용한 SWHAT(Shifted window hierarchical attention) 기법을 제안한다. 이로써, 로컬 window 정보와 전역 정보를 보다 효과적으로 교환할 수 있다.

2. 배 경

2.1. FasterViT

FasterViT 구조는 그림 2와 같이 4단계로 구성되어 있으며, 1, 2단계에서는 convolution 연산을 진행하여 해상도를 감소시키고 채널 수를 증가시킨다. 3, 4단계에서는 HAT(Hierarchical attention) layer를 사용하여 전체 특징 맵을 공간적으로 추론한다. 본 논문에서는 3단계의 HAT에 SWHAT 기법을 적용한다.

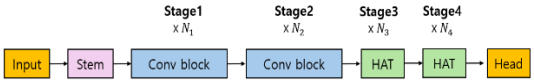

그림 2. FasterViT의 전체 구조도

2.2. 캐리어 토큰

그림 3과 같이 FasterViT HAT layer 들어가기 앞서, 각 window마다 캐리어 토큰을 생성한다. 캐리어 토큰의 역할은 해당하는 로컬 window attention과 모든 캐리어 토큰과의 attention 연산을 통해 지역 정보 및 전역적 공간적 정보를 모두 포착하는 것이다.

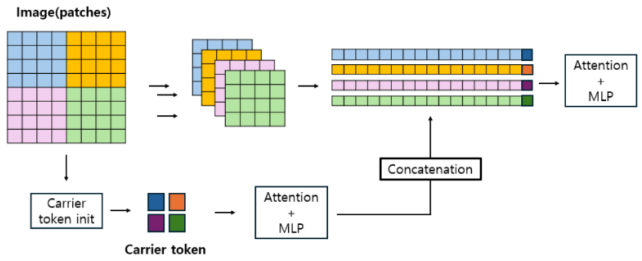

그림 3. HAT layer 일부

캐리어 토큰(X_{ct})은 각 window를 식 2와 같이 pooling하여 $L = 2^c$개로 array 형태로 생성한다. 이 과정을 캐리어 토큰 초기화라고 하며, 하나의 로컬 window의 요약을 나타낸다. 또한, 식 1과 같이 convolution 연산을 통해 캐리어 토큰의 위치 인코딩을 진행한다.

$$\widehat{X_c} = Conv_{3x3}(x) \quad (1)$$

$$\widehat{X_{ct}} = AvgPool_{H^2 \to n^2 L}(\widehat{X_c}) \quad (2)$$

3. 본 론

3.1 SWHAT

기존 FasterViT에서 채택하고 있는 window 분할 방식은 식 3과 같이, 입력 이미지 H를 $M \times M$ 로컬 window로 나눈다. ($M = \frac{H^2}{k^2}, k : window\ size$)

$$\widehat{x1} = Split_{k \times k}(x) \quad (3)$$

본 논문에서는 window 간 정보를 더 효율적으로 공유하기 위해 기존 $M \times M$ window 분할 방식 이외에 Swin transformer의 window shift을 활용한 SWHAT를 제안한다. SWHAT는 이전 layer의 window 위치에서 ($\frac{2}{k}$, $\frac{2}{k}$) 픽셀만큼 shift 시킨 후 window 분할 과정을 진행한다.

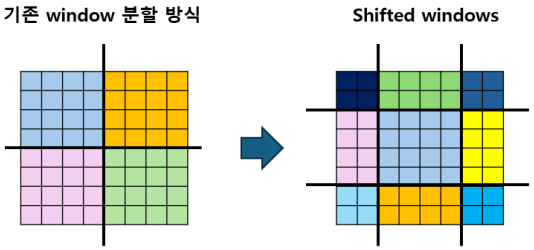

그림 4. SWHAT 기법 예시

3.2 캐리어 토큰 초기화 및 upsampling

FasterViT 3단계의 HAT 블록은 그림 5 (a)와 같이 한 번 생성된 캐리어 토큰으로 N번 반복 진행한다. 그리고 마지막 layer에서 캐리어 토큰을 upsampling하여 캐리어 토큰의 정보를 다음 입력에 반영해준다. 그러나 본 논문에서는 그림 5 (b)에서 2가지 window 분할 기법을 번갈아 가며 FasterViT 3단계를 수행한다. 매 layer마다 Window의 분할 기법이 변경된다. 그림 4를 예시로, 기존 4개 window에서 9개의 window로 변경되면서 캐리어 토큰을 그대로 사용할 수가 없는 문제점이 있다. 그래서 본 논문에서는 그림 6과 같이 HAT에서 사용했던 캐리어 토큰을 upsampling하여 다음 SWHAT 입력에 정보를 반영해준다. 그 뒤에 shift된 window 분할에 맞춰 새로운 캐리어 토큰을 다시 생성한다. 이렇게 진행함으로써 정확도 개선을 가져왔다.

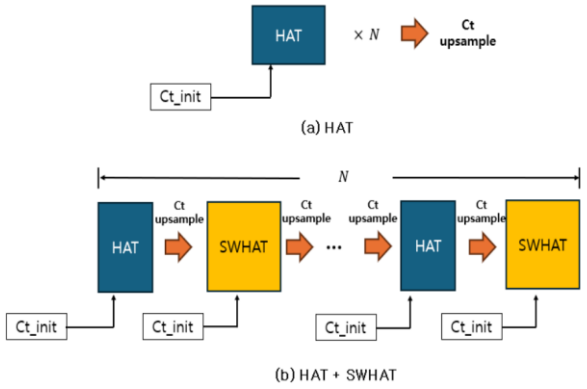

그림 5. 3단계에서의 HAT와 HAT+SWHAT 진행 방식

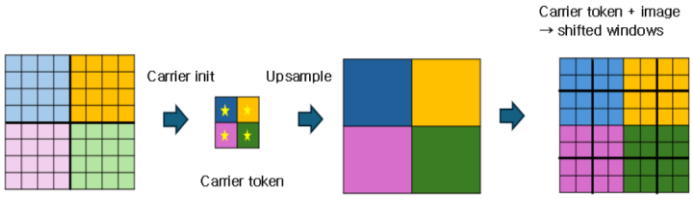

그림 6. 캐리어 토큰 upsampling 과정 및 window shift 분할

4. 실 험

4.1 실험 환경

본 논문에서는 SWHAT 기법의 성능을 평가하기 위해 ImageNet-1k dataset을 사용하였다. 베이스 라인 모델은 FasterViT-0을 사용하였고, 모델의 정확도는 ImageNet-1k의 테스트 세트를 통해 이미지 분류 작업에서 측정하였다. 실험 환경은 Nvidia RTX 3090에서 진행되었다. 배치 사이즈는 1024, 학습률은 1.25e-3으로 설정하였다.

4.2 실험 방법

SWHAT 기법의 성능을 비교하기 위해 베이스 라인 모델인 FasterViT-0에서 SWHAT layer의 적용 유무에 따른 이미지 분류 정확도를 비교하였다. FasterViT 구조의 3단계만 변경하여 실험하였다.

4.3 실험 결과 및 분석

표 1과 같이 SWHAT layer의 적용 유무에 따른 이미지 분류 작업의 정확도 성능을 비교하였다.

표 1. 이미지 분류 정확도

Method	Accuracy (%)
HAT	81.422
HAT + SWHAT (No upsampling)	81.664
HAT + SWHAT (Upsampling)	81.694

HAT+SWHAT layer를 적용한 모델이 기존 방식 HAT보다 0.272% 향상된 것을 볼 수 있다. SWHAT 기법은 window를 shift시켜 window 간의 다양한 정보를 교환하게 하였다. 그리고 HAT+SWHAT 실험의 캐리어 토큰 upsampling 유무에 따라 정확도를 측정했다. Upsampling을 적용 시, 정확도 0.030% 향상을 보였다. 이 경우 매 layer마다 window가 shift하면서도 캐리어 토큰 정보를 보존하여 정확도를 개선했다.

또한, 기존 HAT와 SWHAT를 매 layer마다 번갈아 진행했을 때, window 정보 교환이 더 효과적으로 일어났다. 표 2의 실험결과는 한 단계 내에서 매 layer마다 번갈아 진행했을 때와 HAT layer를 N/2번 진행 후 SWHAT layer N/2번 진행을 적용한 결과를 비교하였다. 번갈아 진행하는 방식이 정확도가 0.156% 더 높았다. 이를 통해 매 layer마다 번갈아 진행하는 것이 정보 교환이 잘 일어나는 것을 알 수 있었다.

표 2. SWHAT 진행 방식 변경

Method	Accuracy (%)
HAT× N/2 + SWHAT × N/2	81.538
(HAT + SWHAT) N/2	81.694

표 3의 실험결과는 HAT와 HAT+SWHAT의 throughput(images/s)를 나타내고 있다. HAT+SWHAT 기법은 HAT와 비슷한 throughput을 보여주고 있다.

표 3. HAT와 HAT+SWHAT의 throughput 비교

Batch size	128	256	512	1024
HAT	438.48/s	708.33/s	702.39/s	677.29/s
HAT+SWHAT	446.31/s	707.12/s	696.56/s	670.72/s

실험 결과를 통해 SWHAT는 HAT보다 정확도 개선을 달성하고, 비슷한 throughput을 만족하였다.

5. 결 론

본 논문에서는 FasterViT의 HAT layer 기반으로 Swin transformer의 shifted window 기법을 활용하여 SWHAT 기법을 제안하였다.

이는 FasterViT의 고정된 window 분할 방식으로 완벽한 장거리 의존성 정보를 포착하지 못하는 문제점을 해결하였다. 또한, HAT layer와 SWHAT layer를 번갈아 가면서 진행하기 때문에, 캐리어 토큰을 매 layer마다 생성해줘야 한다. 이로 인해 이전 layer의 캐리어 토큰의 정보가 손실된다. 그래서 캐리어 토큰의 정보를 보존할 수 있도록 캐리어 토큰을 upsampling 작업을 진행하여 정확도 성능을 높였다.

실험을 통해 HAT+SWHAT 기법이 기존보다 정확도 측면에서 0.272% 향상을 보였다. 또한, 매 layer마다 HAT와 shifted windows HAT를 번갈아 진행하여 정보 교환을 다양하게 하는 효과를 보았다.

참 고 문 헌

[1] Vaswani, et al. “Attention is all you need.” Advances in neural information processing systems 30. (2017)

[2] Alexey Dosovitskiy, et al. “An image is worth 16x16 words: Transformers for image recognition at scale.”, In international conference on Learning representations. (2020)

[3] Hugo Touvron, et al. “Training data-efficient image transformers & distillation through attention.”, In International Conference on Machine Learning, pp. 10347-10357.(2021)

[4] Ze Liu, et al. “Swin transformer: Hierarchical vision transformer using shifted windows.”, In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012-10022. (2021)

[5] Ali Hatamizadeh, et al. “Fastervit: Fast vision transformers with hierarchical attention.”, International Conference on Learning representations. (2024)